

STAT 600 - HW 5

Kevin Korsurat

All Rcpp/RcppArmadillo can be found in my [GitHub](#).

(a)

Below is the derivation of the likelihood function, $p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha)$.

$$\begin{aligned}
p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) &= \prod_{j=1}^{112} P(X_j|\lambda_1, \lambda_2, \alpha) \\
&= \left[\prod_{j=1}^{\theta} \frac{e^{-\lambda_1} \lambda_1^{X_j}}{X_j!} \right] \left[\prod_{j=\theta+1}^{112} \frac{e^{-\lambda_2} \lambda_2^{X_j}}{X_j!} \right] \\
&= \frac{\lambda_1^{\sum_{j=1}^{\theta} X_j} e^{-\lambda_1 \theta} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_2 (112-\theta)}}{\prod_{j=1}^{112} X_j!} \\
&\propto \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2 (112-\theta)}
\end{aligned}$$

In order to perform Gibbs sampler, we need four conditional probabilities derived below.

$$\begin{aligned}
p(\theta|\lambda_1, \lambda_2, \alpha, \mathbf{X}) &\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\theta) p(\lambda_1|\alpha) p(\lambda_2|\alpha) p(\alpha) \\
&\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\theta) \\
&= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2 (112-\theta)} \frac{1}{111} \mathbb{I}_{\theta \in \{1, 2, \dots, 111\}} \\
&= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\theta(\lambda_1 - \lambda_2)} \mathbb{I}_{\theta \in \{1, 2, \dots, 111\}} \\
&= \left(\frac{\lambda_1}{\lambda_2} \right)^{\sum_{j=1}^{\theta} X_j} e^{-\theta(\lambda_1 - \lambda_2)} \mathbb{I}_{\theta \in \{1, 2, \dots, 111\}}
\end{aligned}$$

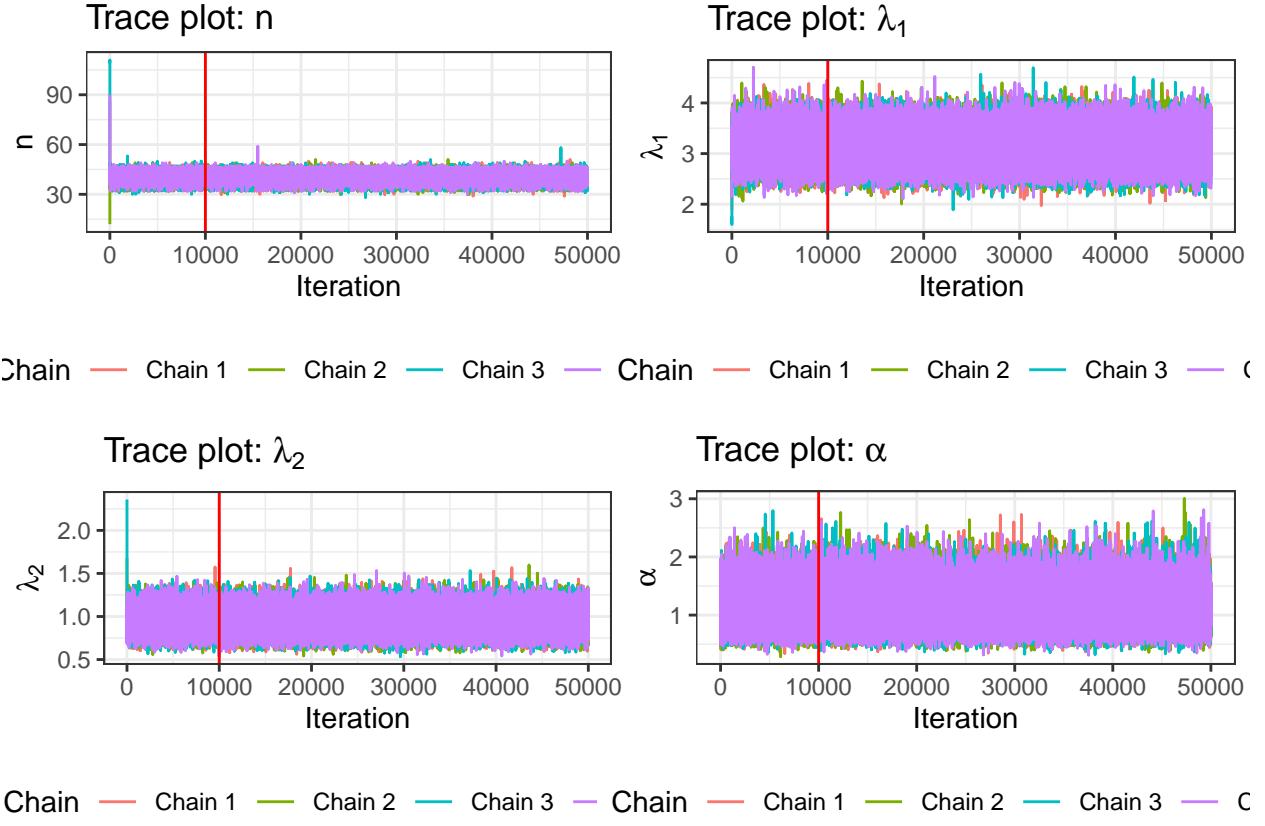
$$\begin{aligned}
p(\lambda_1|\theta, \lambda_2, \alpha, \mathbf{X}) &\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\theta) p(\lambda_1|\alpha) p(\lambda_2|\alpha) p(\alpha) \\
&\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\lambda_1|\alpha) \\
&= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2 (112-\theta)} \frac{\alpha^3}{\Gamma(3)} \lambda_1^2 e^{-\alpha \lambda_1} \\
&\propto \lambda_1^{\sum_{j=1}^{\theta} X_j} e^{-\lambda_1 \theta} \lambda_1^2 e^{-\alpha \lambda_1} \\
&\propto \lambda_1^{2 + \sum_{j=1}^{\theta} X_j} e^{-\lambda_1 (\theta + \alpha)} \\
&\equiv \text{Gamma} \left(3 + \sum_{j=1}^{\theta} X_j, \theta + \alpha \right)
\end{aligned}$$

$$\begin{aligned}
p(\lambda_2|\theta, \lambda_1, \alpha, \mathbf{X}) &\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\theta) p(\lambda_1|\alpha) p(\lambda_2|\alpha) p(\alpha) \\
&\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\lambda_2|\alpha) \\
&= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2(112-\theta)} \frac{\alpha^3}{\Gamma(3)} \lambda_2^2 e^{-\alpha \lambda_2} \\
&\propto \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_2(112-\theta)} \lambda_2^2 e^{-\alpha \lambda_2} \\
&\propto \lambda_2^{2 + \sum_{j=\theta+1}^{112} X_j} e^{-\lambda_2(112-\theta+\alpha)} \\
&\equiv \text{Gamma}\left(3 + \sum_{j=\theta+1}^{112} X_j, 112 - \theta + \alpha\right)
\end{aligned}$$

$$\begin{aligned}
p(\alpha|\theta, \lambda_1, \lambda_2, \mathbf{X}) &\propto p(\mathbf{X}|\theta, \lambda_1, \lambda_2, \alpha) p(\theta) p(\lambda_1|\alpha) p(\lambda_2|\alpha) p(\alpha) \\
&\propto p(\lambda_1|\alpha) p(\lambda_2|\alpha) p(\alpha) \\
&= \frac{\alpha^3}{\Gamma(3)} \lambda_1^2 e^{-\alpha \lambda_1} \frac{\alpha^3}{\Gamma(3)} \lambda_2^2 e^{-\alpha \lambda_2} \frac{10}{\Gamma(10)} \alpha^9 e^{-10\alpha} \\
&\propto \alpha^{15} e^{-\alpha(10 + \lambda_1 + \lambda_2)} \\
&\equiv \text{Gamma}(16, 10 + \lambda_1 + \lambda_2)
\end{aligned}$$

(b)

I will run the model for 50,000 iterations while letting the first 10,000 iterations serve as burn-in. Below are the trace plots for each parameter.



According to the result, we see that all parameters are mixing well in all 4 MCMC chains. This result can be confirmed by the Gelman-Rubin statistics (R_c). The result shows that the R_c for all chains, along with the Multivariate PSRF, is equal to 1, indicating that the MCMC results are converged.

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1
## [2,]      1      1
## [3,]      1      1
## [4,]      1      1
##
## Multivariate psrf
##
## 1
```

The autocorrelation for each chain is shown in the tables below. The results illustrate that there is no autocorrelation for all parameters in all 4 chains. In conclusion, the MCMC is mixing well based on the provided diagnostic results.

Table 1: Autocorrelation of Chain 1 of the MCMC when using gamma distribution as a prior

	n	lambda 1	lambda 2	alpha
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000

	n	lambda 1	lambda 2	alpha
Lag 1	0.1205445	0.0757074	0.0665339	0.0065350
Lag 5	0.0038037	0.0031725	0.0103407	0.0008376
Lag 10	0.0027811	-0.0041621	-0.0048952	0.0024916
Lag 100	0.0016483	-0.0043413	0.0001637	-0.0055586
Lag 250	0.0016625	0.0011437	-0.0023011	0.0051191

Table 2: Autocorrelation of Chain 2 of the MCMC when using gamma distribution as a prior

	n	lambda 1	lambda 2	alpha
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.1160043	0.0766880	0.0615223	0.0096173
Lag 5	-0.0006337	-0.0000961	-0.0005727	-0.0008149
Lag 10	0.0049091	0.0027595	0.0019397	0.0062104
Lag 100	0.0048860	0.0006930	0.0051027	0.0016174
Lag 250	-0.0043204	-0.0029495	-0.0047723	-0.0052680

Table 3: Autocorrelation of Chain 3 of the MCMC when using gamma distribution as a prior

	n	lambda 1	lambda 2	alpha
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.1367481	0.0799255	0.0632148	0.0105693
Lag 5	0.0039829	-0.0003998	-0.0010688	-0.0018549
Lag 10	0.0020779	0.0005267	-0.0019302	-0.0037925
Lag 100	-0.0002401	0.0002901	0.0058188	-0.0000857
Lag 250	0.0006807	0.0016806	-0.0073944	0.0008575

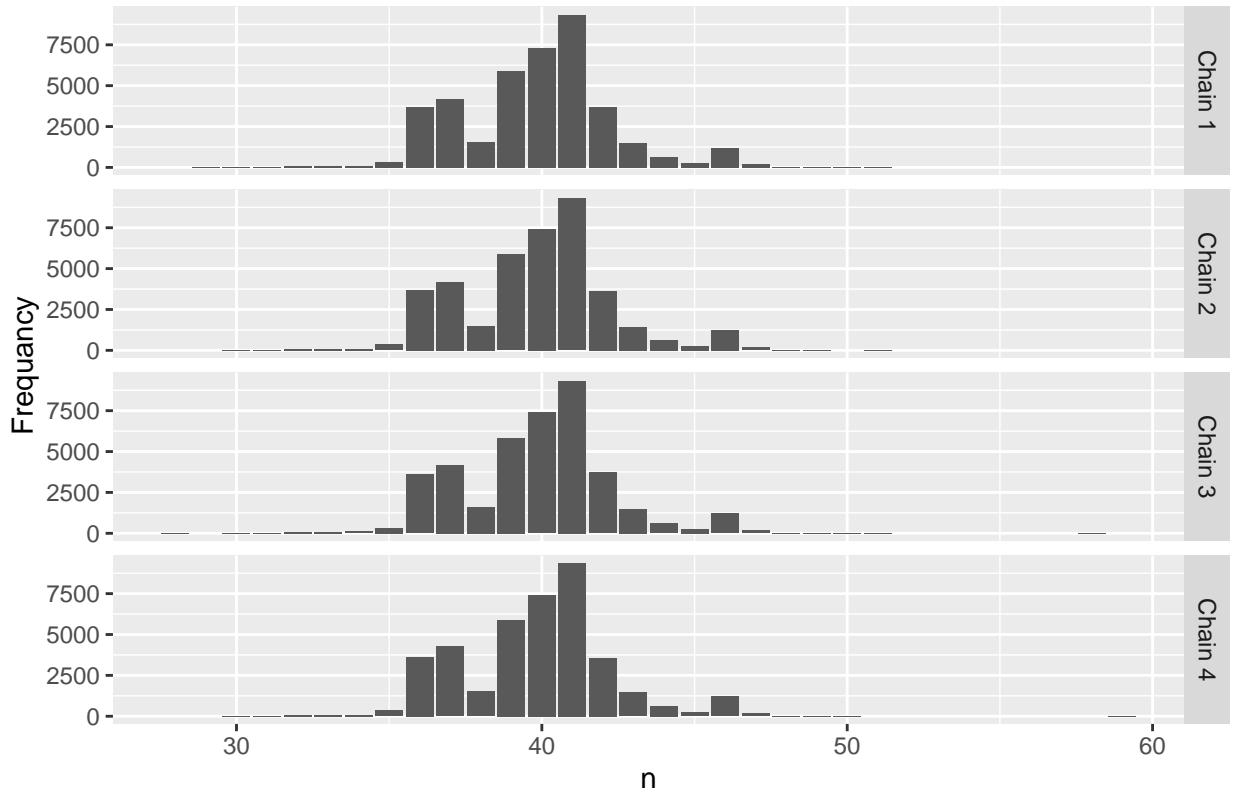
Table 4: Autocorrelation of Chain 4 of the MCMC when using gamma distribution as a prior

	n	lambda 1	lambda 2	alpha
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.1139266	0.0763294	0.0673175	0.0084810
Lag 5	-0.0083673	0.0033349	0.0023812	0.0006595
Lag 10	0.0013873	-0.0010623	-0.0008751	0.0012229
Lag 100	0.0023954	0.0048464	-0.0024079	-0.0045887
Lag 250	0.0074166	-0.0005359	-0.0070417	0.0048522

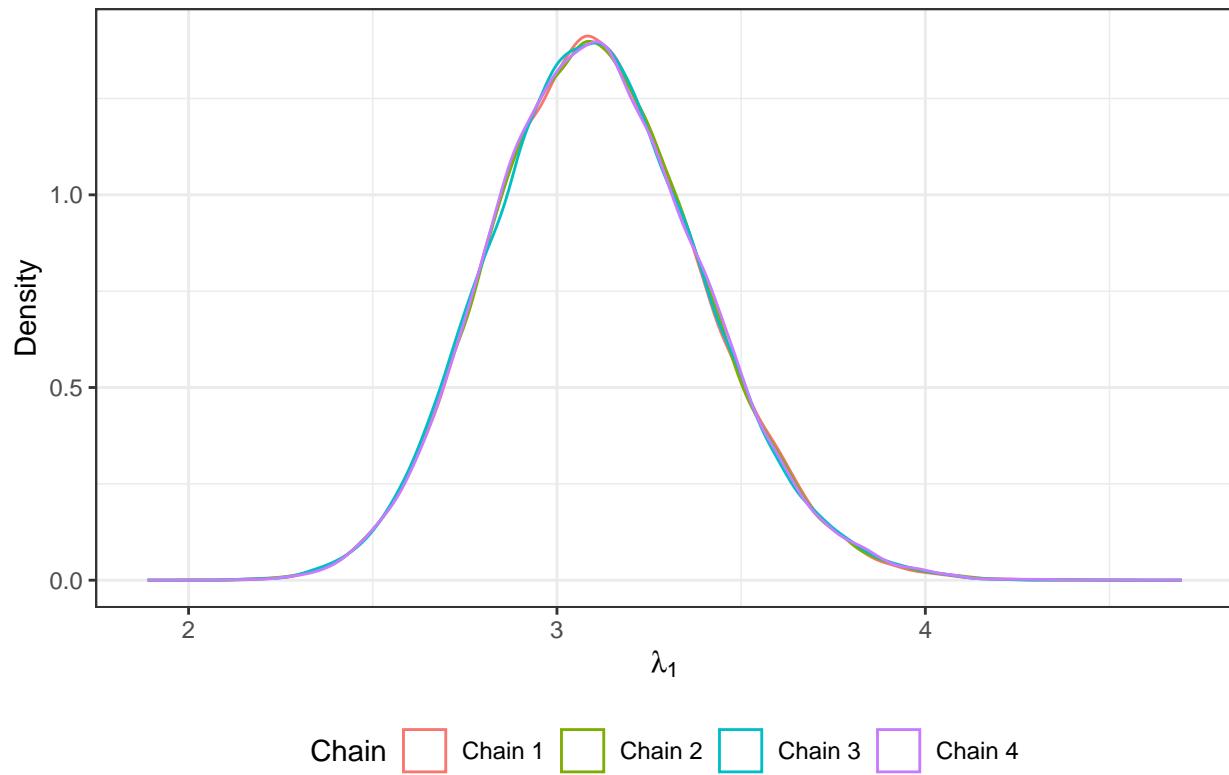
(c)

Below are the density histogram plots for all parameters in all 4 chains.

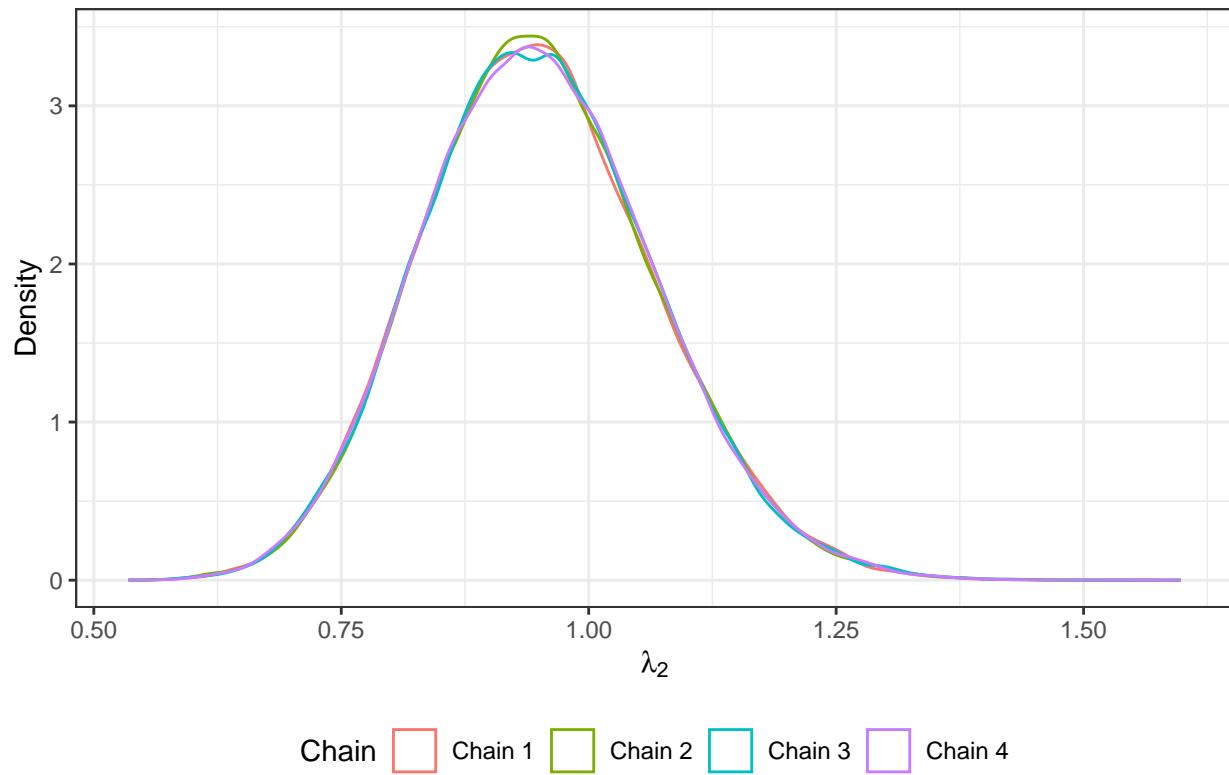
Distribution of n



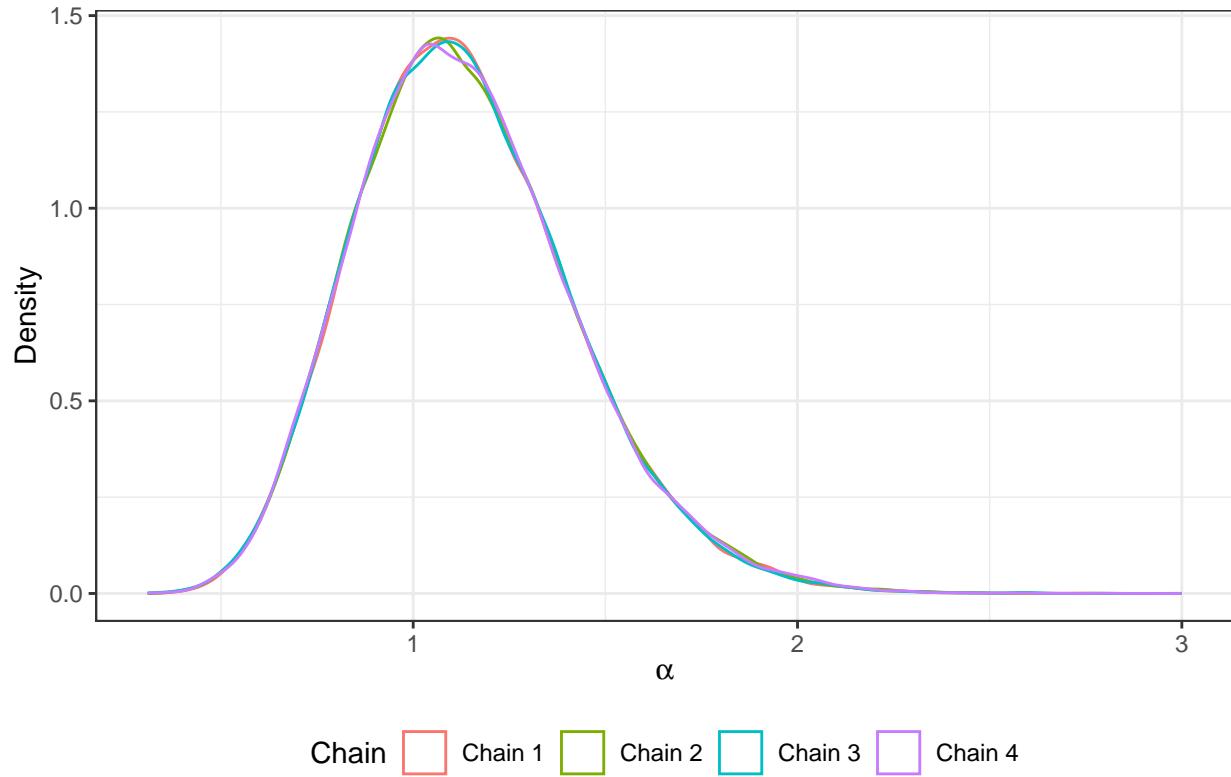
Density plot: λ_1



Density plot: λ_2



Density plot: α



The table below shows the mean and the 95% Highest Posterior Density (HPD) Interval for all parameters in all 4 chains.

	Chain 1	Chain 2	Chain 3	Chain 4
n	39.8492 (35, 44)	39.8456 (34, 44)	39.8574 (35, 44)	39.8415 (35, 44)
ld 1	3.1097 (2.5627, 3.6803)	3.1101 (2.5678, 3.6875)	3.1088 (2.5494, 3.6771)	3.1113 (2.5412, 3.6648)
ld 2	0.9495 (0.7242, 1.1826)	0.9503 (0.7303, 1.1873)	0.9502 (0.7208, 1.1781)	0.9501 (0.7272, 1.1855)
alpha	1.1389 (0.6027, 1.6964)	1.1406 (0.6099, 1.713)	1.1377 (0.6145, 1.708)	1.1398 (0.6136, 1.7201)

For the continuous random variables (λ_1 , λ_2 , and α), I believe using the HPD interval is reasonable because it captures the highest density regions of the posterior distribution, providing a concise summary of parameter uncertainty. However, issues may arise when applying a 95% HPD interval to the discrete random variable (n) as it may not accurately represent the discrete nature of the distribution, leading to potentially misleading interval estimates.

(d)

In the context of this analysis, the parameters λ_1 and λ_2 represent the Poisson distribution parameters for two distinct groups. As highlighted in part (c), λ_1 exceeds λ_2 , implying that the mean and standard deviation of disasters were higher during the initial stages of the coal-mining era compared to later periods.

This trend aligns with the observed data, which depicts a decreasing trend in disaster occurrences over time. Moreover, the dispersion in the latter half of the era appears to decrease, with no more than one disaster per year, contrasting with the substantial number of disasters during the early stages.

Transitioning to the parameter θ , we can interpret it as the threshold that delineates the data into two segments. Conceptually, these segments represent distinct eras: one characterized by a high frequency of disasters predating technological advancements, and the other indicating a period marked by the implementation of safety measures to mitigate disasters. With the average value of θ hovering around 40, we can reasonably infer that 1891 signifies the onset of the era where such safety measures were introduced, thus leading to a reduction in disaster occurrences.

(e)

For this question, I will let the prior for λ_i to be the half-Normal distribution. Specifically, $p(\lambda_i | \sigma_i^2) = \frac{\sqrt{2}}{\sqrt{\pi\sigma_i^2}} e^{-\frac{\lambda_i^2}{2\sigma_i^2}} \mathbb{I}_{\lambda_i > 0}$ for $i = 1, 2$.

The conditional probability for θ still be the same as in part (a). Below are the derivation for the conditional probability for λ_i .

$$\begin{aligned} p(\lambda_1 | \theta, \lambda_2, \mathbf{X}) &\propto p(\mathbf{X} | \theta, \lambda_1, \lambda_2) p(\theta) p(\lambda_1) p(\lambda_2) \\ &\propto p(\mathbf{X} | \theta, \lambda_1, \lambda_2) p(\lambda_1) \\ &= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2(112-\theta)} \frac{\sqrt{2}}{\sqrt{\pi\sigma_1^2}} e^{-\frac{\lambda_1^2}{2\sigma_1^2}} \\ &\propto \lambda_1^{\sum_{j=1}^{\theta} X_j} e^{-\lambda_1 \theta - \frac{\lambda_1^2}{2\sigma_1^2}} \\ p(\lambda_2 | \theta, \lambda_1, \mathbf{X}) &\propto p(\mathbf{X} | \theta, \lambda_1, \lambda_2) p(\theta) p(\lambda_1) p(\lambda_2) \\ &\propto p(\mathbf{X} | \theta, \lambda_1, \lambda_2) p(\lambda_2) \\ &= \lambda_1^{\sum_{j=1}^{\theta} X_j} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_1 \theta} e^{-\lambda_2(112-\theta)} \frac{\sqrt{2}}{\sqrt{\pi\sigma_2^2}} e^{-\frac{\lambda_2^2}{2\sigma_2^2}} \\ &\propto \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-\lambda_2(112-\theta) - \frac{\lambda_2^2}{2\sigma_2^2}} \end{aligned}$$

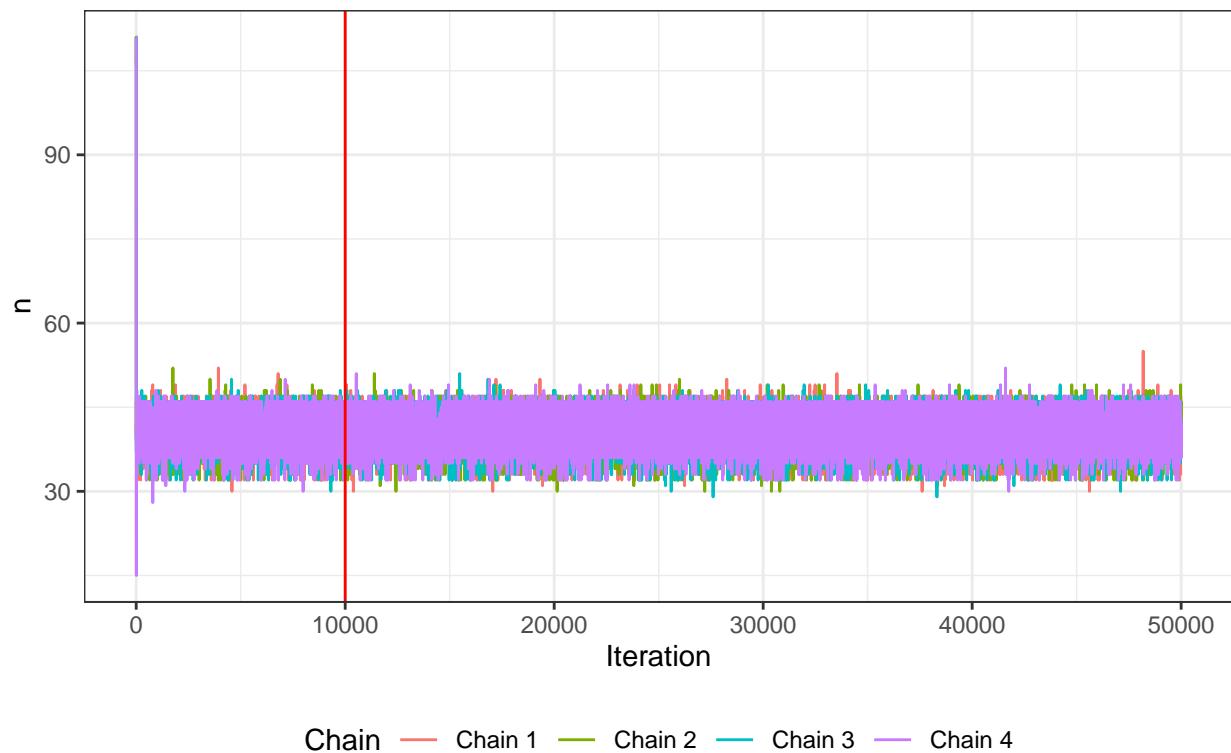
(f)

Since we does not have the closed form for the conditional probability of λ_1 , and λ_2 . I will apply the MH algorithm while letting the proposal distribution, $q(\lambda_i^* | \lambda_i)$, to be Gamma($\lambda_i, 1$). In terms of σ_1^2 and σ_2^2 , the hyperparameters, I will try three sets listed below:

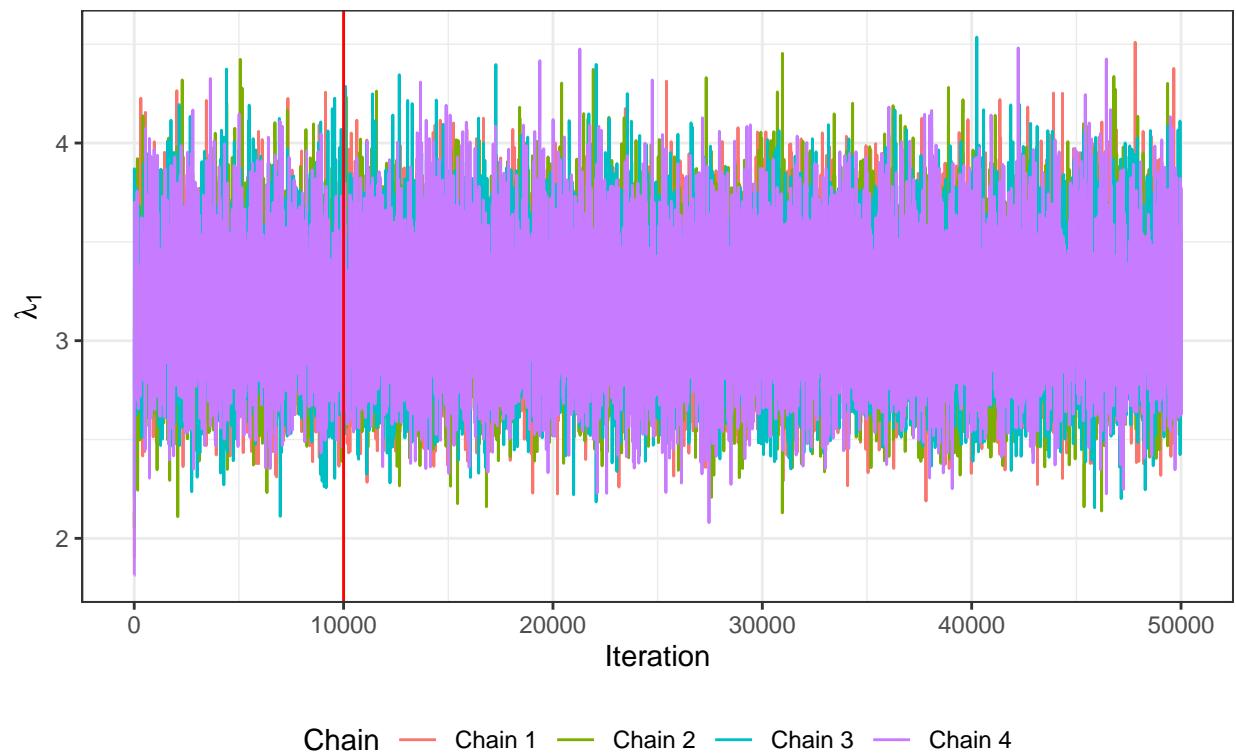
- Non-informative: $\sigma_1^2 = \sigma_2^2 = 1000$
- Non-informative (with less disperse): $\sigma_1^2 = \sigma_2^2 = 10$
- Informative: $\sigma_1^2 = 10$ and $\sigma_2^2 = 1$ as seen in parts (c) and (d), indicate that in more recent years, there are fewer disasters, leading to the conclusion that the number of disasters is less dispersed in the latter years.

Begin with the result from the first case, $\sigma_1^2 = \sigma_2^2 = 1000$.

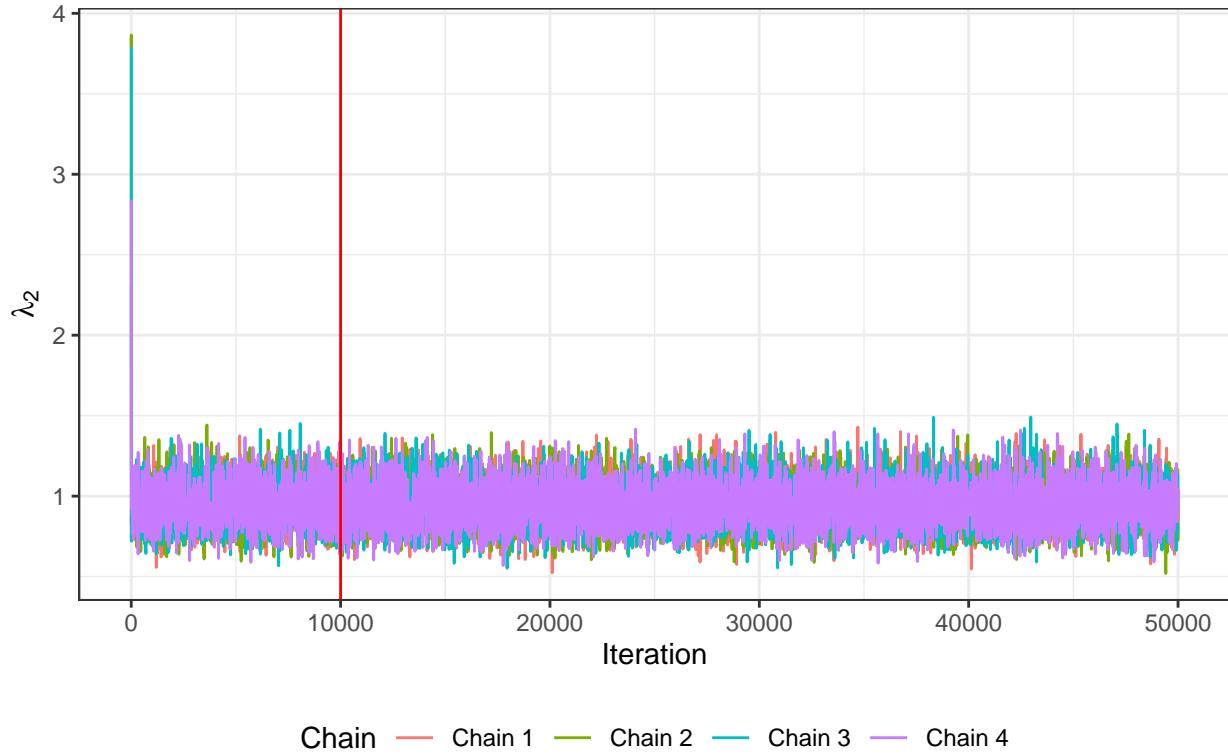
Trace plot: n with $\sigma_1^2 = \sigma_2^2 = 1000$



Trace plot: λ_1 with $\sigma_1^2 = \sigma_2^2 = 1000$



Trace plot: λ_2 with $\sigma_1^2 = \sigma_2^2 = 1000$



```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## [1,]      1      1
## [2,]      1      1
## [3,]      1      1
##
## Multivariate psrf
##
## 1
```

Table 6: Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior ($s_1 = s_2 = 1000$)

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1493293	0.7889060	0.8347368
Lag 5	0.0671020	0.3096653	0.4132971
Lag 10	0.0266993	0.0943487	0.1782897
Lag 100	0.0064071	0.0058409	0.0034237
Lag 250	-0.0004484	-0.0000173	0.0120577

Table 7: Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1000$)

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1487021	0.7892576	0.8393153
Lag 5	0.0502291	0.3079433	0.4373230
Lag 10	0.0188033	0.0990057	0.2025195
Lag 100	0.0004574	-0.0149711	-0.0067645
Lag 250	-0.0039697	-0.0111110	-0.0087003

Table 8: Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1000$)

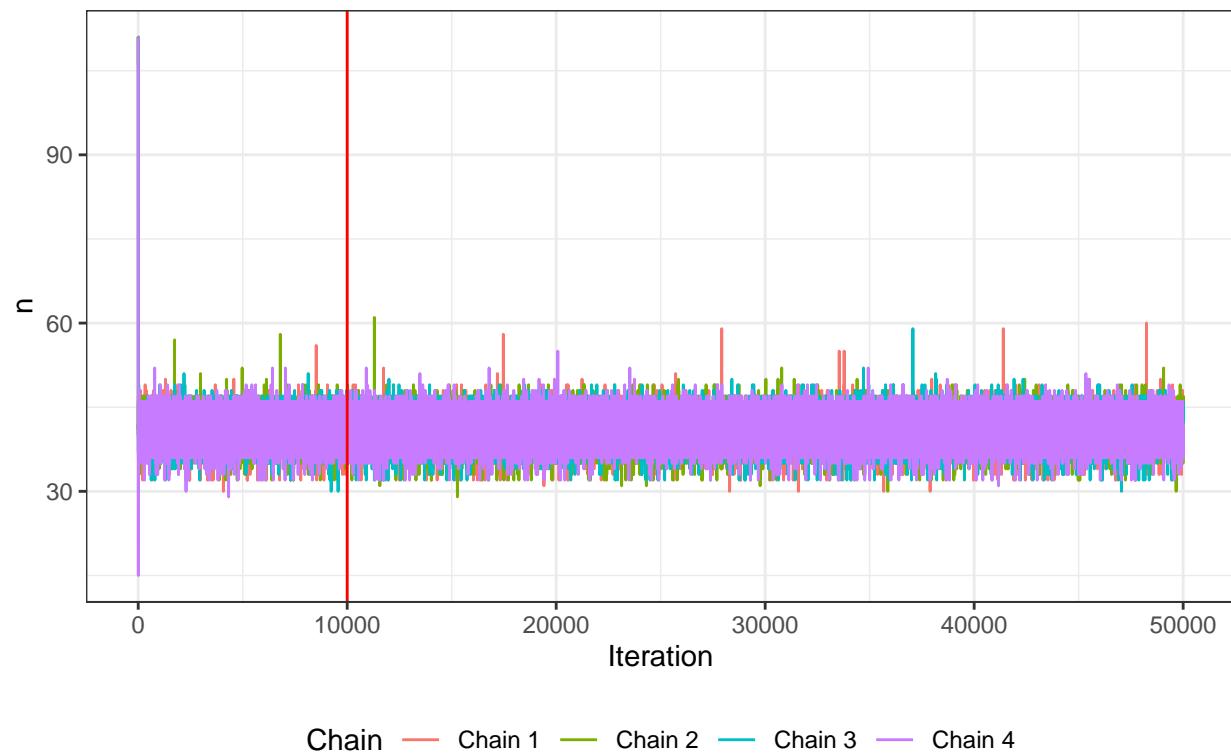
	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1642793	0.7870182	0.8332325
Lag 5	0.0547338	0.3046859	0.4019830
Lag 10	0.0278210	0.0960481	0.1787543
Lag 100	-0.0017881	0.0032771	-0.0143158
Lag 250	0.0029815	-0.0029294	0.0071918

Table 9: Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1000$)

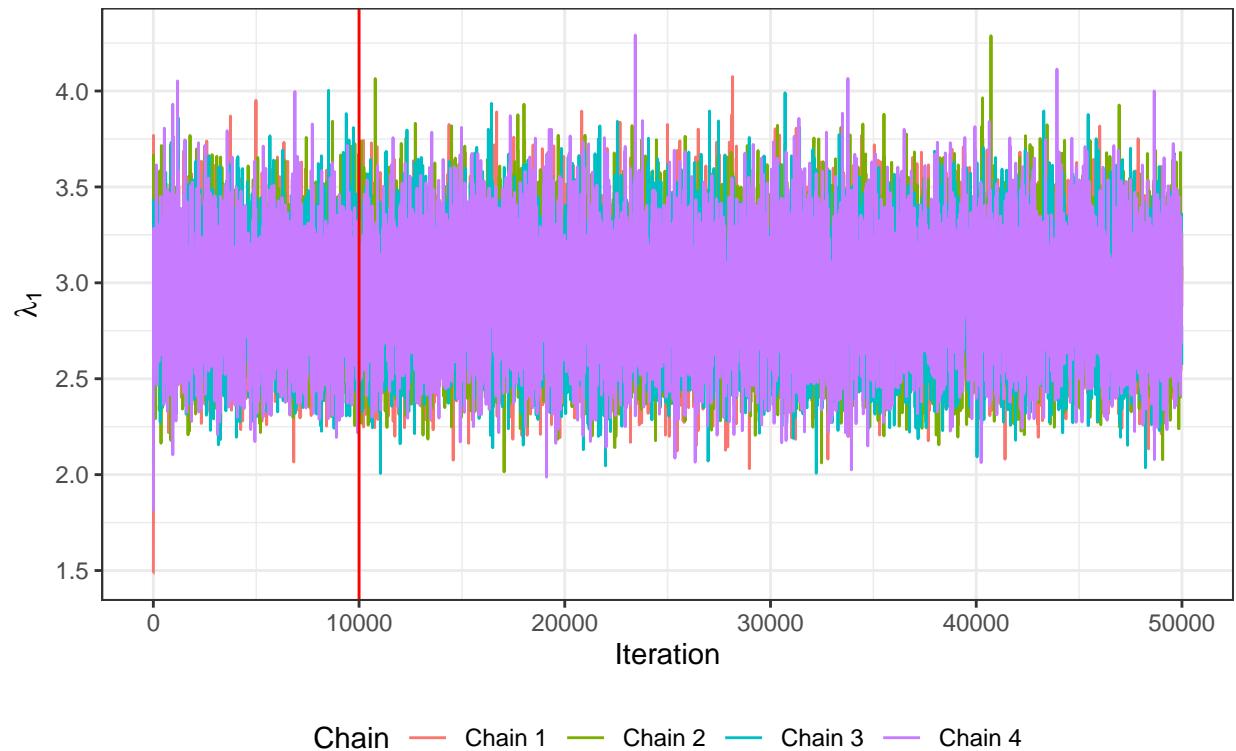
	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1461567	0.7863075	0.8391010
Lag 5	0.0601907	0.3030113	0.4212950
Lag 10	0.0215703	0.0862768	0.1778070
Lag 100	-0.0060932	-0.0082646	-0.0201877
Lag 250	-0.0010751	-0.0083835	0.0110282

Below are the result from the second case, $\sigma_1^2 = \sigma_2^2 = 1$.

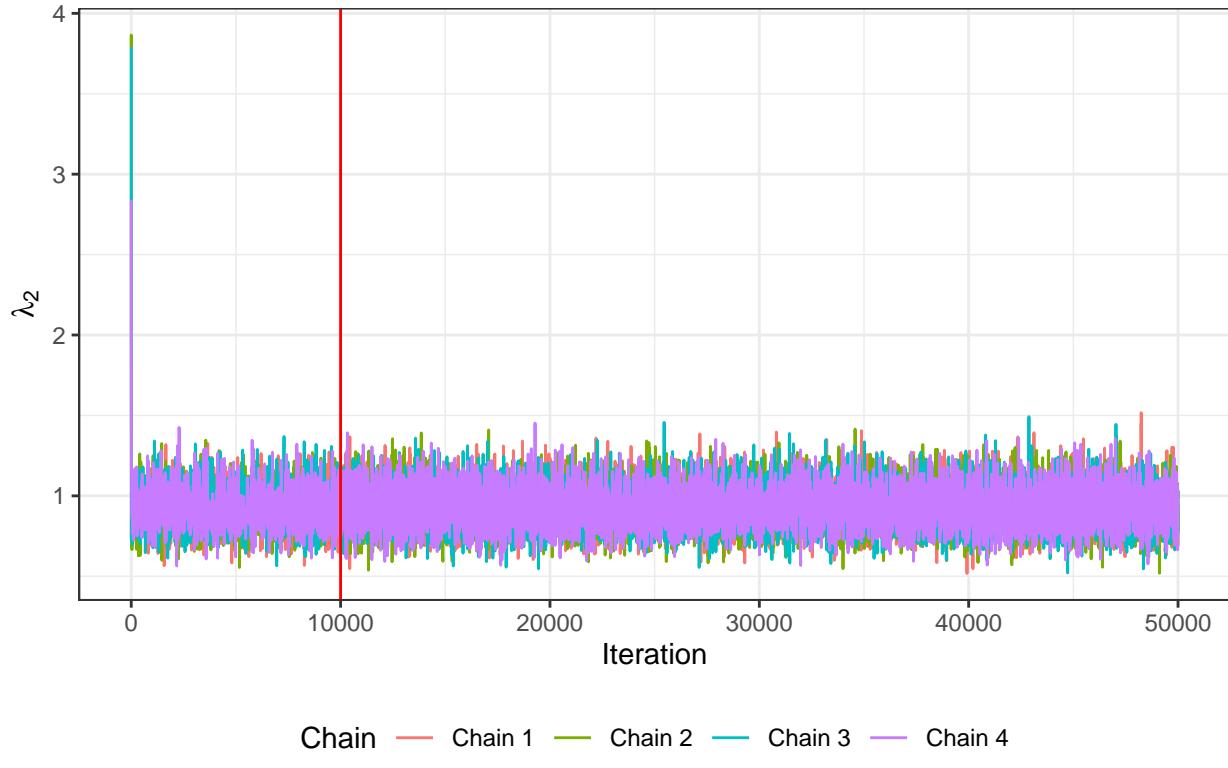
Trace plot: n with $\sigma_1^2 = \sigma_2^2 = 1$



Trace plot: λ_1 with $\sigma_1^2 = \sigma_2^2 = 1$



Trace plot: λ_2 with $\sigma_1^2 = \sigma_2^2 = 1$



Chain — Chain 1 — Chain 2 — Chain 3 — Chain 4

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1
## [2,]      1      1
## [3,]      1      1
##
## Multivariate psrf
##
## 1
```

Table 10: Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior ($s_1 = s_2 = 1$)

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1120849	0.7951744	0.8374155
Lag 5	0.0477130	0.3126982	0.4124920
Lag 10	0.0205374	0.0901444	0.1764021
Lag 100	0.0099967	-0.0009994	0.0098161
Lag 250	-0.0083117	0.0043961	-0.0263306

Table 11: Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1$)

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1226292	0.8051695	0.8367755
Lag 5	0.0507630	0.3402782	0.4272414
Lag 10	0.0248593	0.1240717	0.2094497
Lag 100	-0.0061643	0.0036293	0.0045964
Lag 250	0.0021696	0.0013483	-0.0084563

Table 12: Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1$)

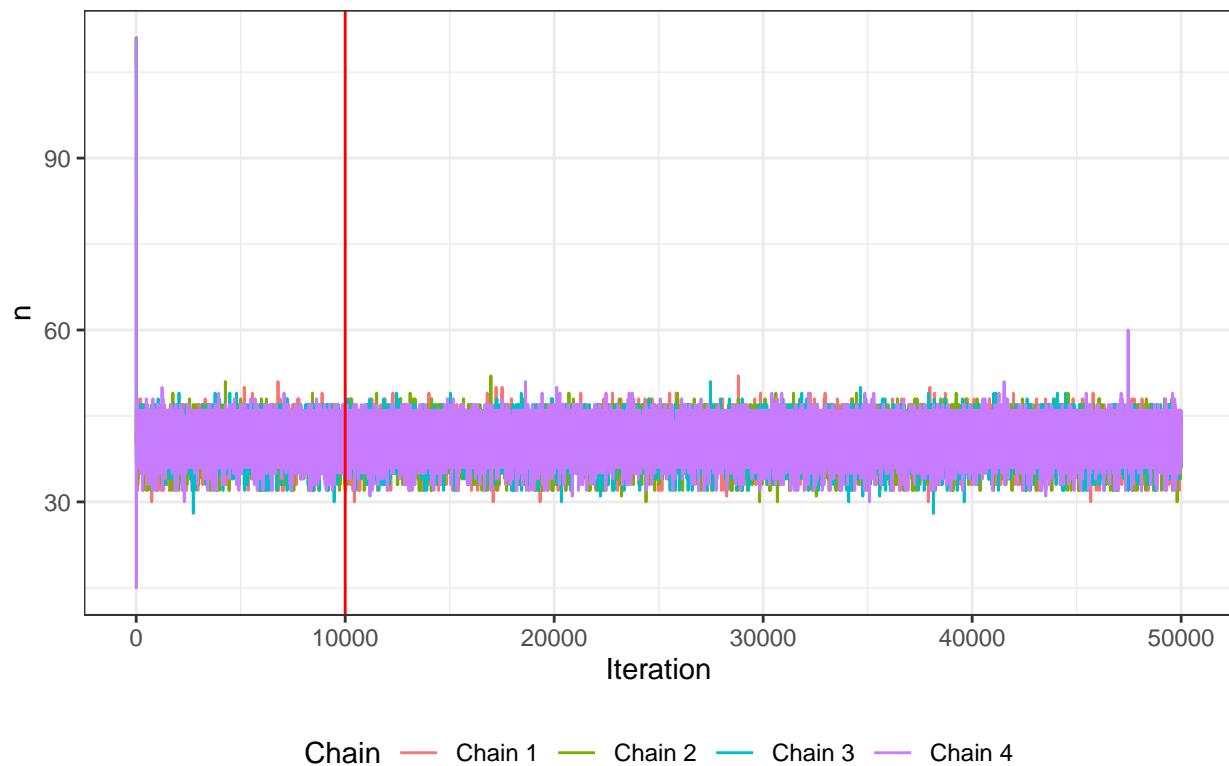
	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1417229	0.7845918	0.8397397
Lag 5	0.0481594	0.3084885	0.4208973
Lag 10	0.0166427	0.1102061	0.1909769
Lag 100	0.0035954	0.0081134	-0.0078235
Lag 250	0.0003585	0.0096808	0.0083507

Table 13: Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior ($s1 = s2 = 1$)

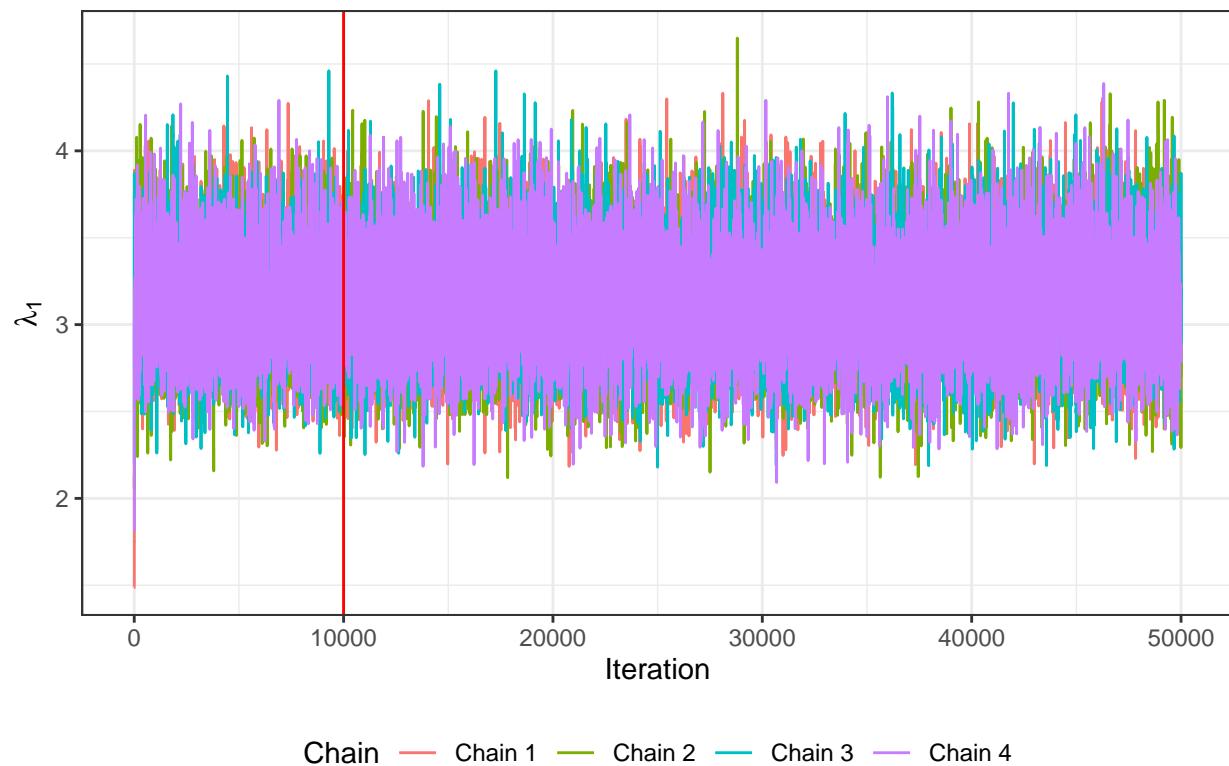
	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1276176	0.8010241	0.8356968
Lag 5	0.0563381	0.3356506	0.4216889
Lag 10	0.0184833	0.1186490	0.1906126
Lag 100	0.0023525	0.0311742	-0.0022488
Lag 250	0.0012585	-0.0006979	-0.0104130

Last, we will use the informative prior, $\sigma_1^2 = 10, \sigma_2^2 = 1$.

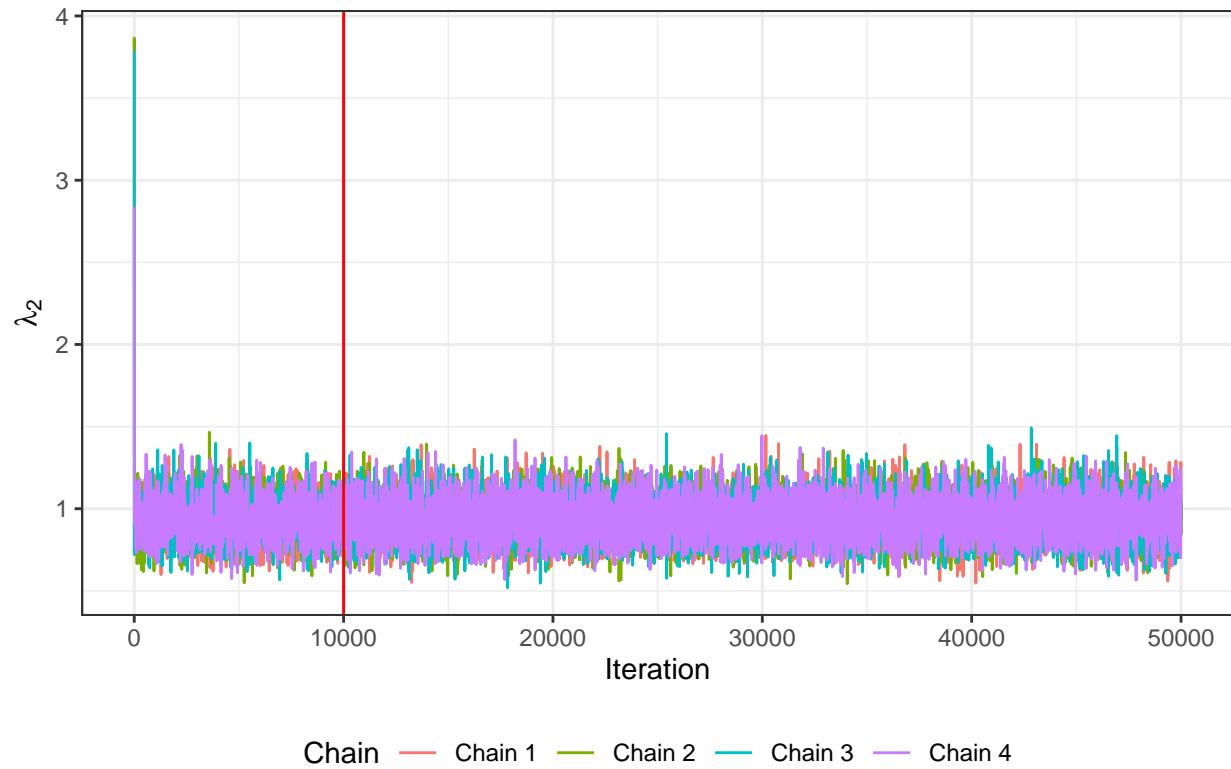
Trace plot: n with the informative prior



Trace plot: λ_1 with the informative prior



Trace plot: λ_2 with the informative prior



```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]      1      1
## [2,]      1      1
## [3,]      1      1
##
## Multivariate psrf
##
## 1
```

Table 14: Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior with the informative prior

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1267260	0.7837624	0.8418066
Lag 5	0.0568999	0.2978595	0.4283822
Lag 10	0.0188207	0.0876225	0.1817375
Lag 100	-0.0020353	0.0130023	-0.0141039
Lag 250	0.0028383	-0.0063137	-0.0073782

Table 15: Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior with the informative prior

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1405894	0.7871021	0.8395107
Lag 5	0.0551072	0.3138291	0.4294673
Lag 10	0.0182668	0.0977282	0.1950892
Lag 100	0.0003689	-0.0016471	-0.0046676
Lag 250	0.0022561	-0.0100135	-0.0105436

Table 16: Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior with the informative prior

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1532401	0.7847709	0.8383674
Lag 5	0.0605840	0.3049321	0.4176690
Lag 10	0.0289717	0.1052002	0.1929306
Lag 100	-0.0083265	-0.0069510	-0.0187748
Lag 250	0.0035479	0.0111425	-0.0029395

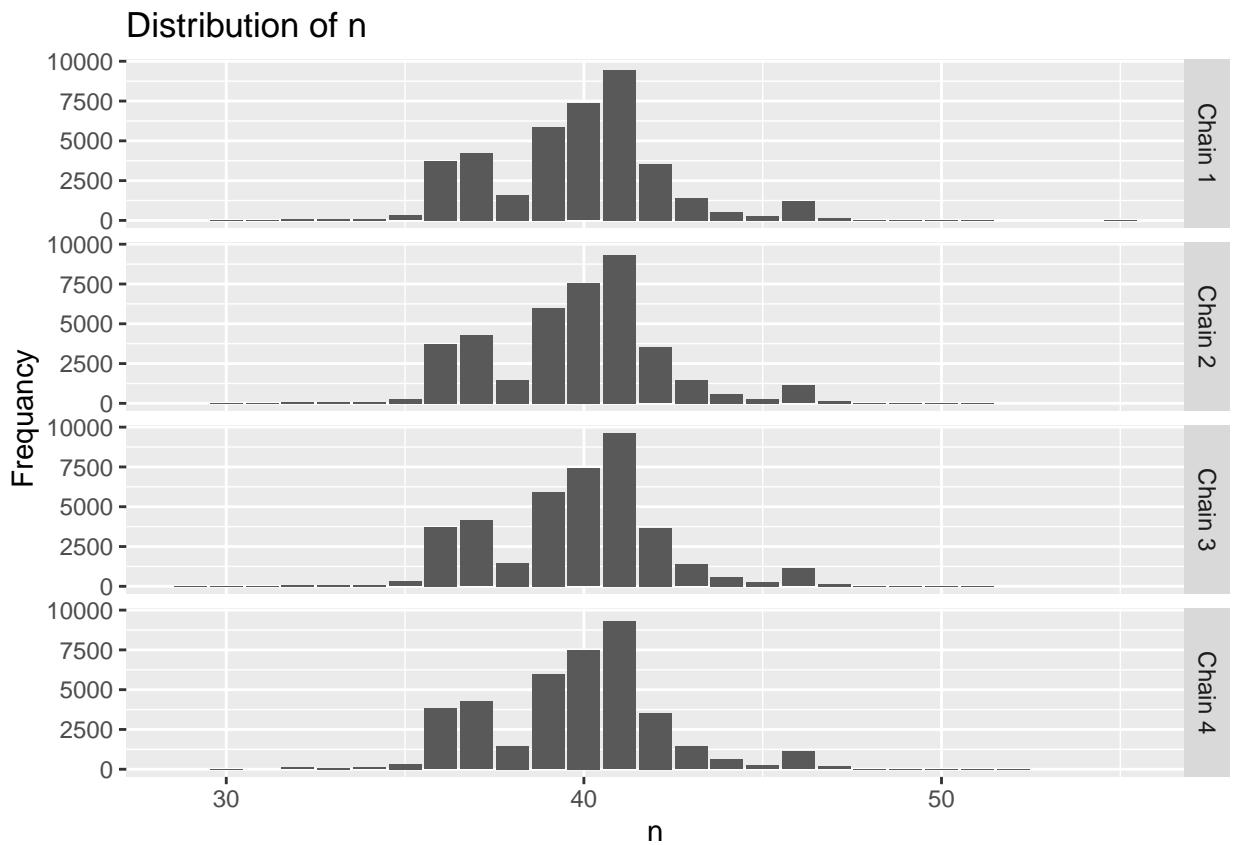
Table 17: Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior with the informative prior

	n	lambda 1	lambda 2
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.1493988	0.7885008	0.8378440
Lag 5	0.0489599	0.3016088	0.4207277
Lag 10	0.0178768	0.0910464	0.1821319
Lag 100	-0.0020428	0.0053819	-0.0000374
Lag 250	-0.0076595	-0.0092905	-0.0104666

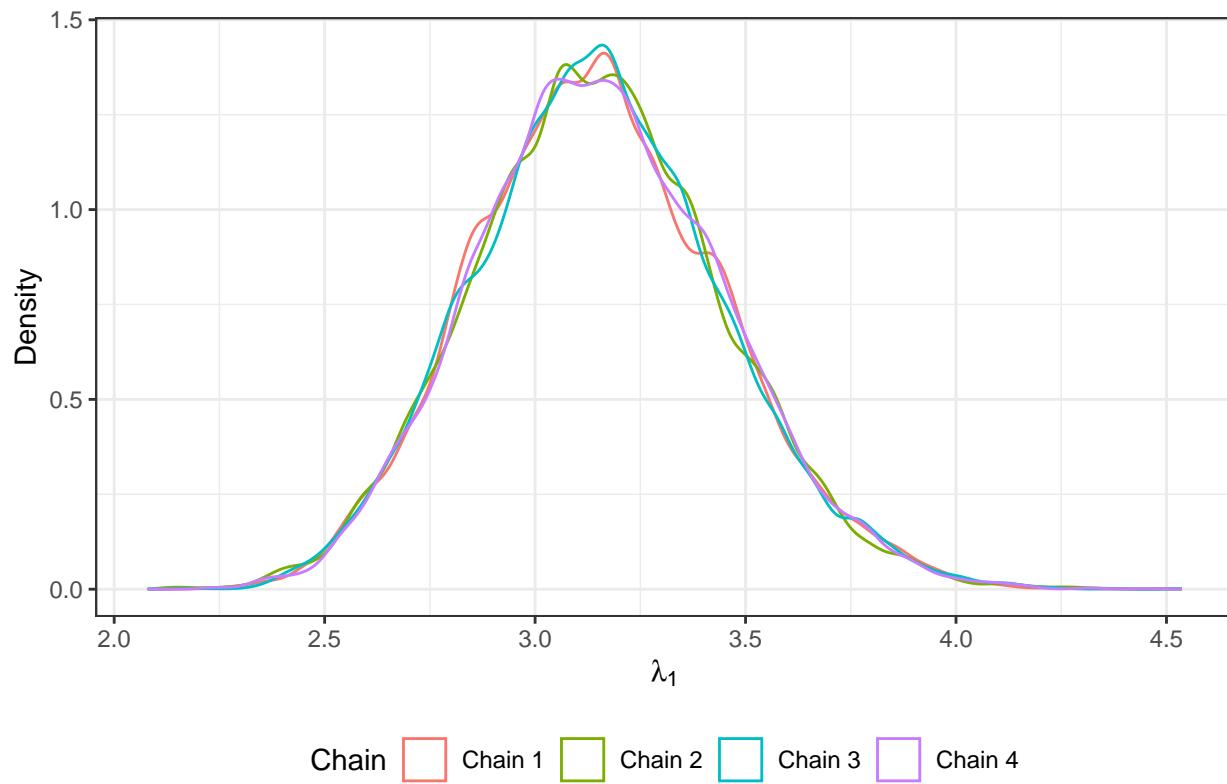
Since the mixing for all sets of hyperparameters is similar (trace plots show convergence in all parameters, $R_c = 1$, and no autocorrelation), I would prefer using the non-informative prior. Therefore, I will proceed with using the first case ($s^2 = 1000$) to address the following questions.

(g)

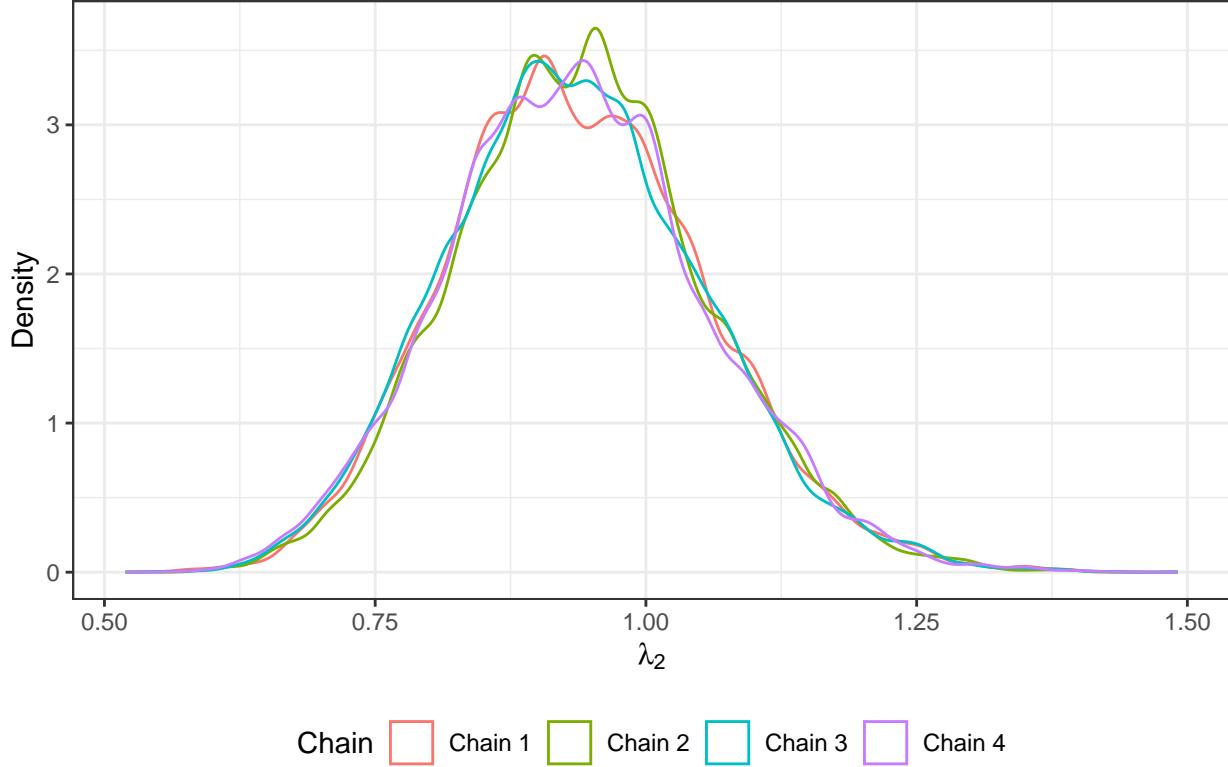
Below are the inference for the parameters when using the Half-Normal as a prior for λ_1 and λ_2 . I will proceed by looking at the plots.



Density plot: λ_1



Density plot: λ_2



Followed by the HPD interval for each parameter.

	Chain 1	Chain 2	Chain 3	Chain 4
n	39.8209 (35, 44)	39.8082 (35, 44)	39.8274 (35, 44)	39.7973 (35, 44)
ld 1	3.1501 (2.5619, 3.7206)	3.1496 (2.556, 3.6994)	3.1514 (2.5496, 3.71)	3.155 (2.589, 3.7354)
ld 2	0.9377 (0.7101, 1.1724)	0.9416 (0.7291, 1.1784)	0.9352 (0.6989, 1.1626)	0.9363 (0.6973, 1.1627)

According to the results, we might notice that the Half-Normal prior gives slightly higher estimates in all parameters compared to the Gamma prior. The trace plots for both priors look similar to each other. However, the density plots for λ_1 and λ_2 when using the Gamma prior are smoother than those for the Half-Normal prior.

While we can set σ^2 to be extremely high to allow the model to explore the whole parameter space when using the Half-Normal prior, I would still prefer using the model with the Gamma prior as we can sample each parameter directly from the known distribution. Using the Half-Normal in this case prevents us from having a closed-form solution to the posterior distribution.

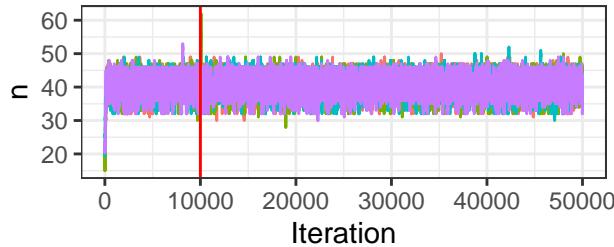
(h)

```
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
```

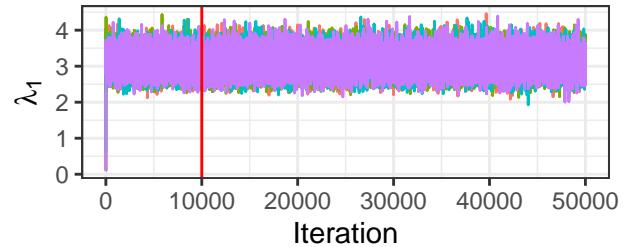
```
## |-----|
## |-----|-----|
## |-----|-----|
```

We begin by running the first model (Gamma prior) with Nimble. We might notice that the average computational time is 4.2432 seconds, slightly slower than when using Rcpp (0.9332 seconds). Another issue we might want to consider is that we can run the model in parallel if we use Rcpp. However, Nimble does not allow this, which means it will take longer to finish all chains in the long run.

Trace plot: n (Gamma Prior with Nir)

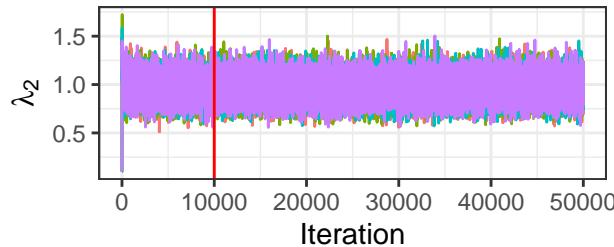


Trace plot: λ_1 (Gamma Prior with Ni

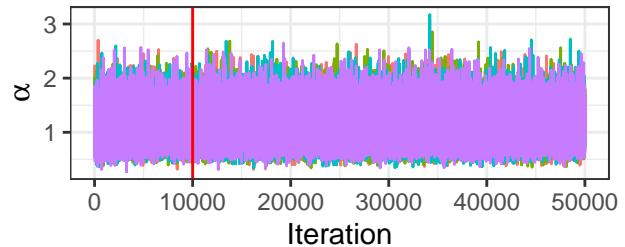


Chain — Chain 1 — Chain 2 — Chain 3 — Chain — Chain 1 — Chain 2 — Chain 3 — C

Trace plot: λ_2 (Gamma Prior with N



Trace plot: α (Gamma Prior with Nir



Chain — Chain 1 — Chain 2 — Chain 3 — Chain — Chain 1 — Chain 2 — Chain 3 — C

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## alpha            1      1
## lambda1         1      1
## lambda2         1      1
## theta           1      1
##
## Multivariate psrf
##
## 1
```

Table 19: Autocorrelation of Chain 1 of the MCMC when using Gamma prior with Nimble

	alpha	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.0066239	0.6623040	0.6704938	0.7534562
Lag 5	0.0097433	0.1535604	0.1545232	0.2898196
Lag 10	0.0054261	0.0329516	0.0427108	0.1241707
Lag 100	0.0036913	-0.0041924	0.0078930	0.0095994
Lag 250	-0.0056928	-0.0000877	0.0027812	0.0062757

Table 20: Autocorrelation of Chain 2 of the MCMC when using Gamma prior with Nimble

	alpha	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.0022525	0.6521887	0.6657531	0.7563991
Lag 5	0.0041523	0.1402673	0.1655287	0.2959176
Lag 10	-0.0066830	0.0130442	0.0661432	0.1292248
Lag 100	-0.0023327	0.0010642	0.0058872	0.0024303
Lag 250	-0.0021703	-0.0145774	0.0037895	-0.0152399

Table 21: Autocorrelation of Chain 3 of the MCMC when using Gamma prior with Nimble

	alpha	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.0103988	0.6641841	0.6538207	0.7381048
Lag 5	0.0081851	0.1554957	0.1270307	0.2529046
Lag 10	0.0089447	0.0282459	0.0121568	0.0832466
Lag 100	0.0044062	0.0119831	0.0008180	0.0233368
Lag 250	0.0076133	0.0047706	-0.0029771	0.0161821

Table 22: Autocorrelation of Chain 4 of the MCMC when using Gamma prior with Nimble

	alpha	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000	1.0000000
Lag 1	0.0073169	0.6564481	0.6596746	0.7490605
Lag 5	-0.0023515	0.1380515	0.1511652	0.2859019
Lag 10	-0.0043158	0.0286049	0.0250525	0.1155861
Lag 100	-0.0007021	0.0007445	-0.0163390	-0.0047283
Lag 250	0.0059425	0.0078195	0.0056006	-0.0073636

	Chain 1	Chain 2	Chain 3	Chain 4
n	39.8867 (35, 45)	39.8443 (34, 44)	39.9015 (35, 44)	39.8218 (35, 44)

	Chain 1	Chain 2	Chain 3	Chain 4
ld 1	3.11 (2.5668, 3.688)	3.1077 (2.5421, 3.6676)	3.1068 (2.5616, 3.6993)	3.1075 (2.5543, 3.6778)
ld 2	0.9493 (0.7253, 1.181)	0.9505 (0.7314, 1.1903)	0.9482 (0.7282, 1.1769)	0.9523 (0.733, 1.1848)
alpha	1.1373 (0.612, 1.7011)	1.1394 (0.6083, 1.7103)	1.1387 (0.615, 1.7154)	1.1373 (0.6042, 1.7022)

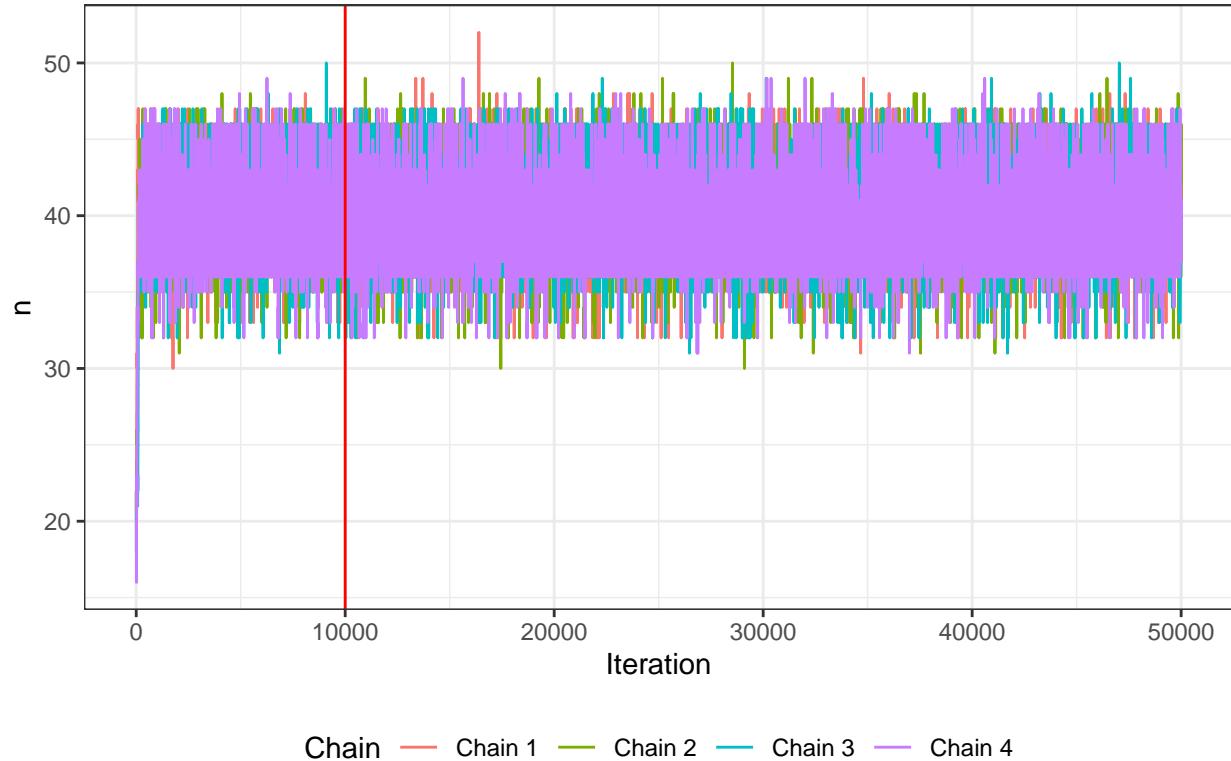
According to the results, we notice that both Nimble and Rcpp yield similar trace plots and estimates. However, in terms of autocorrelation, Rcpp performs better as the autocorrelation decreases faster compared to Nimble. In terms of R_c , both methods give the same result ($R_c = 1$)

Then, we will run the second model (the Half-Normal prior) with Nimble.

```
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
## |-----|-----|-----|-----|
```

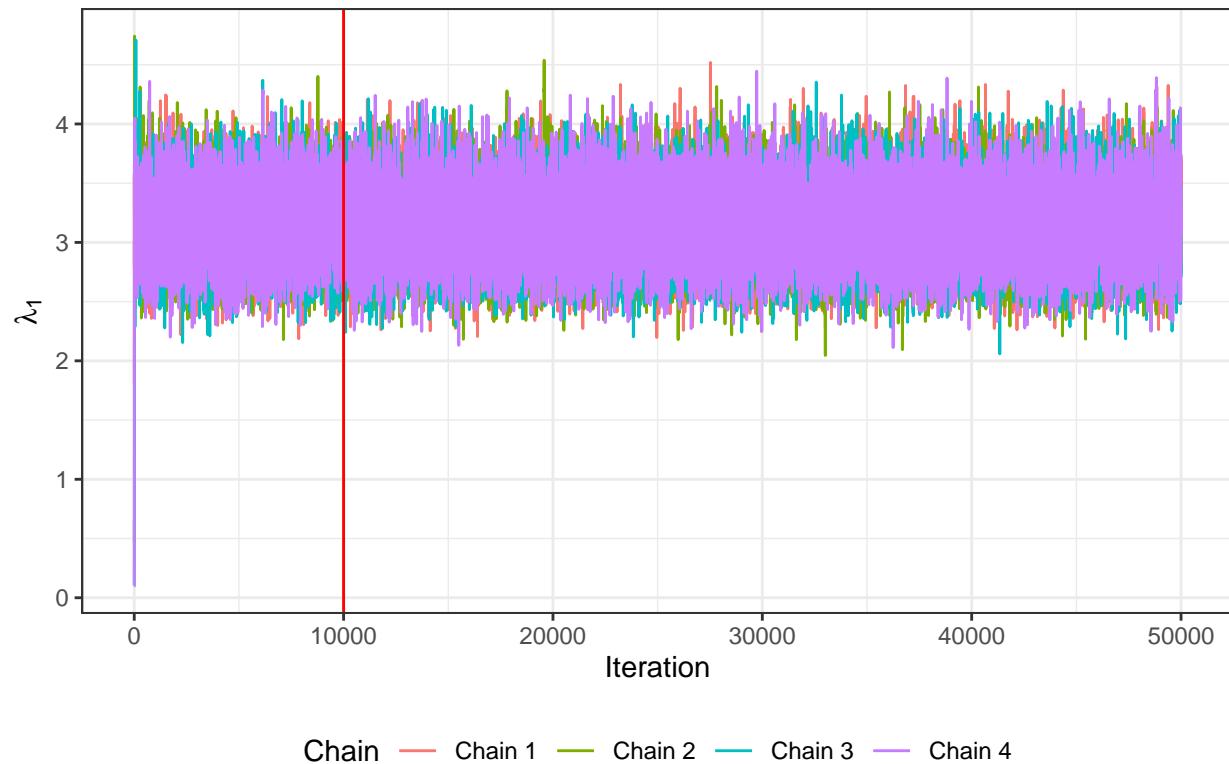
Similarly, we might notice that the average computational time is 3.5631 seconds, slightly slower than when using Rcpp (0.4917 seconds).

Trace plot: n (HN Prior with Nimble)

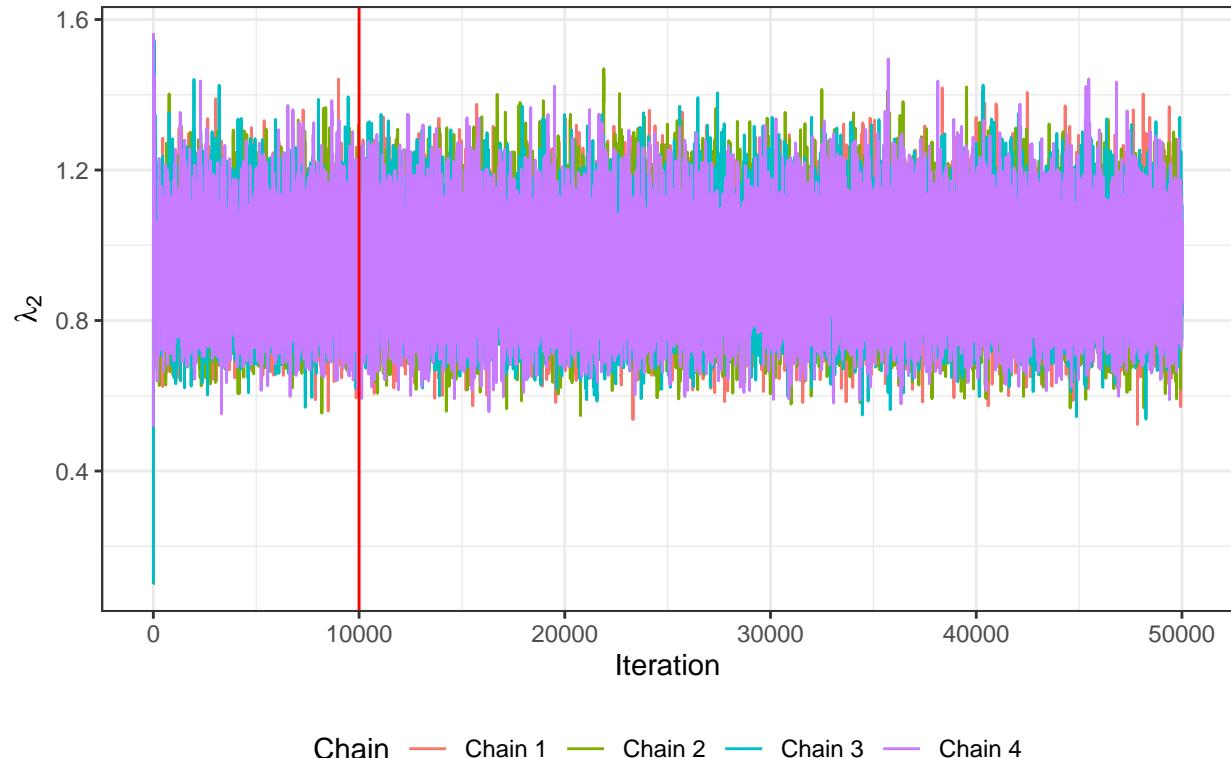


Chain — Chain 1 — Chain 2 — Chain 3 — Chain 4

Trace plot: λ_1 (HN Prior with Nimble)



Trace plot: λ_2 (HN Prior with Nimble)



Chain — Chain 1 — Chain 2 — Chain 3 — Chain 4

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## lambda1         1         1
## lambda2         1         1
## theta           1         1
##
## Multivariate psrf
##
## 1
```

Table 24: Autocorrelation of Chain 1 of the MCMC when using HN prior with Nimble

	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.6658924	0.6607735	0.7412507
Lag 5	0.1477379	0.1545412	0.2619847
Lag 10	0.0357665	0.0296015	0.0903793
Lag 100	-0.0013893	-0.0065380	-0.0118852
Lag 250	-0.0015870	0.0018259	-0.0044473

Table 25: Autocorrelation of Chain 2 of the MCMC when using HN prior with Nimble

	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.6635854	0.6597494	0.7492025
Lag 5	0.1578655	0.1469624	0.2795039
Lag 10	0.0200187	0.0290834	0.0976312
Lag 100	0.0063454	-0.0159657	-0.0031875
Lag 250	0.0090359	-0.0103488	0.0047603

Table 26: Autocorrelation of Chain 3 of the MCMC when using HN prior with Nimble

	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.6590815	0.6591859	0.7623254
Lag 5	0.1656990	0.1537915	0.3078773
Lag 10	0.0297410	0.0542550	0.1387824
Lag 100	-0.0009266	-0.0118279	0.0072713
Lag 250	0.0054240	-0.0022864	-0.0041184

Table 27: Autocorrelation of Chain 4 of the MCMC when using HN prior with Nimble

	lambda 1	lambda 2	n
Lag 0	1.0000000	1.0000000	1.0000000
Lag 1	0.6654951	0.6551876	0.7480176
Lag 5	0.1591993	0.1400558	0.2822327
Lag 10	0.0287817	0.0447444	0.1170566
Lag 100	0.0128570	0.0035984	0.0011094
Lag 250	-0.0082746	0.0103157	-0.0134568

	Chain 1	Chain 2	Chain 3	Chain 4
n	39.8351 (35, 44)	39.8046 (35, 44)	39.804 (35, 44)	39.8151 (35, 44)
ld 1	3.1472 (2.587, 3.7379)	3.1492 (2.5879, 3.7196)	3.1462 (2.5777, 3.7131)	3.1527 (2.5723, 3.7337)
ld 2	0.9373 (0.7121, 1.1668)	0.9395 (0.7138, 1.174)	0.938 (0.7143, 1.1716)	0.9363 (0.7035, 1.1638)

Similarly, both Rcpp and Nimble yield similar results. The only aspect where I believe Rcpp performs better is in managing autocorrelation among iterations.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)

library(tidyverse)
library(knitr)
library(Rcpp)
library(RcppArmadillo)
library(foreach)
library(doParallel)
library(ggplot2)
library(latex2exp)
library(gridExtra)
library(HDInterval)
library(coda)
library(nimble)

path <- "/Users/kevin-imac/Desktop/Github - Repo/"
if(! file.exists(path)){
  path <- "/Users/kevinkvp/Desktop/Github Repo/"
}

sourceCpp(paste0(path, "HW5MCMC/src/main.cpp"))

### Import the data
dat <- read.table(paste0(path, "HW5MCMC/coal.dat"), header = TRUE)

### Run the model (4 chains)
set.seed(5, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
resultGamma <- foreach(t = 1:4) %dopar% {

  start_time <- Sys.time()
  result <- gibbsGamma(iter = 50000, dat = dat$disasters)
  run_time <- difftime(Sys.time(), start_time, units = "secs")

  list(run_time = run_time, result = result)
}

stopImplicitCluster()

### Function: Trace plot from List
ttGGPlot <- function(allChain, burn_in, yLab, titleLab){
  ggplot(allChain, aes(x = iter, y = Estimate, color = Chain)) +
    geom_line() +
    theme_bw() +
    theme(legend.position = "bottom") +
    labs(x = "Iteration", y = yLab, title = titleLab) +
    geom_vline(xintercept = burn_in, color = "red")
}

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){resultGamma[[x]]$result[, 1]}) %>%
```

```

`colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("n"), titleLab = "Trace plot: n")

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){resultGamma[[x]]$result[, 2]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\lambda_1$"), titleLab = TeX("Trace plot: $\\lambda_1$"))

### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){resultGamma[[x]]$result[, 3]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\lambda_2$"), titleLab = TeX("Trace plot: $\\lambda_2$"))

### Trace plot (all chains) for alpha
p4 <- sapply(1:4, function(x){resultGamma[[x]]$result[, 4]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\alpha$"), titleLab = TeX("Trace plot: $\\alpha$"))

grid.arrange(p1, p2, p3, p4)

### Create the MCMC object
mcmcGamma <- mcmc.list(lapply(1:4, function(x){mcmc(resultGamma[[x]]$result, start = 10001)}))
gelman.diag(mcmcGamma)
kable(autocorr.diag(mcmcGamma[[1]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2", "alpha"),
      caption = "Autocorrelation of Chain 1 of the MCMC when using gamma distribution as a prior")

kable(autocorr.diag(mcmcGamma[[2]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2", "alpha"),
      caption = "Autocorrelation of Chain 2 of the MCMC when using gamma distribution as a prior")

kable(autocorr.diag(mcmcGamma[[3]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2", "alpha"),
      caption = "Autocorrelation of Chain 3 of the MCMC when using gamma distribution as a prior")

kable(autocorr.diag(mcmcGamma[[4]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2", "alpha"),
      caption = "Autocorrelation of Chain 4 of the MCMC when using gamma distribution as a prior")

### Plots

```

```

d1 <- sapply(1:4, function(x){resultGamma[[x]]$result[-c(1:10000), 1]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  group_by(Chain, Estimate) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = Estimate, y = n)) +
  geom_bar(stat = "identity") +
  facet_grid(Chain ~ .) +
  labs(title = "Distribution of n", y = "Frequency", x = "n")

dPlot <- function(allChain, xLab, titleLab){
  ggplot(allChain, aes(x = Estimate, color = Chain)) +
    geom_density() +
    theme_bw() +
    theme(legend.position = "bottom") +
    labs(x = xLab, y = "Density", title = titleLab)
}

d2 <- sapply(1:4, function(x){resultGamma[[x]]$result[-c(1:10000), 2]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  dPlot(xLab = TeX("$\\lambda_1$"), titleLab = TeX("Density plot: $\\lambda_1$"))

d3 <- sapply(1:4, function(x){resultGamma[[x]]$result[-c(1:10000), 3]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  dPlot(xLab = TeX("$\\lambda_2$"), titleLab = TeX("Density plot: $\\lambda_2$"))

d4 <- sapply(1:4, function(x){resultGamma[[x]]$result[-c(1:10000), 4]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  dPlot(xLab = TeX("$\\alpha$"), titleLab = TeX("Density plot: $\\alpha$"))

d1
d2
d3
d4

### Function: Mean and SD
meanHDI <- function(x, dplace = 4){
  hdiX <- round(as.numeric(hdi(x)), digits = dplace)
  mm <- round(mean(x), digits = dplace)
  paste0(mm, " (", hdiX[1], ", ", hdiX[2], ")")
}

sapply(1:4, function(x){apply(resultGamma[[x]]$result[-c(1:10000), ], 2, meanHDI)}) %>%
  `rownames<-` (c("n", "ld 1", "ld 2", "alpha")) %>%
  kable(col.names = paste0("Chain ", 1:4))

```

```

### Half-Normal (Non-informative: s2 = 1000)
set.seed(5, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
resultHN1 <- foreach(t = 1:4) %dopar% {

  start_time <- Sys.time()
  result <- gibbsHalfN(iter = 50000, s2_1 = 1000, s2_2 = 1000, dat = dat$disasters)
  run_time <- difftime(Sys.time(), start_time, units = "secs")

  list(run_time = run_time, result = result)

}
stopImplicitCluster()

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){resultHN1[[x]]$result[, 1]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("n"), titleLab = TeX("Trace plot: n with  $\sigma^2$  =  $\sigma^2$ "))

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){resultHN1[[x]]$result[, 2]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\lambda_1$"), titleLab = TeX("Trace plot: $\lambda_1$ with  $\sigma^2$ "))

### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){resultHN1[[x]]$result[, 3]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\lambda_2$"), titleLab = TeX("Trace plot: $\lambda_2$ with  $\sigma^2$ "))

p1
p2
p3

mcmcHN1 <- mcmc.list(lapply(1:4, function(x){mcmc(resultHN1[[x]]$result, start = 10001)}))
gelman.diag(mcmcHN1)

kable(autocorr.diag(mcmcHN1[[1]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior"

kable(autocorr.diag(mcmcHN1[[2]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior"

```

```

kable(autocorr.diag(mcmcHN1[[3]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior

kable(autocorr.diag(mcmcHN1[[4]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior

### Half-Normal (Non-informative: s2 = 1)
set.seed(5, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
resultHN2 <- foreach(t = 1:4) %dopar% {

  start_time <- Sys.time()
  result <- gibbsHalfN(iter = 50000, s2_1 = 1, s2_2 = 1, dat = dat$disasters)
  run_time <- difftime(Sys.time(), start_time, units = "secs")

  list(run_time = run_time, result = result)

}
stopImplicitCluster()

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){resultHN2[[x]]$result[, 1]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("n"), titleLab = TeX("Trace plot: n with  $\sigma^2 = 1$ "))

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){resultHN2[[x]]$result[, 2]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\lambda_1$"), titleLab = TeX("Trace plot: $\lambda_1$ with  $\sigma^2 = 1$ "))

### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){resultHN2[[x]]$result[, 3]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\lambda_2$"), titleLab = TeX("Trace plot: $\lambda_2$ with  $\sigma^2 = 1$ "))

p1
p2
p3

mcmcHN2 <- mcmc.list(lapply(1:4, function(x){mcmc(resultHN2[[x]]$result, start = 10001)}))
gelman.diag(mcmcHN2)

```

```

kable(autocorr.diag(mcmcHN2[[1]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior

kable(autocorr.diag(mcmcHN2[[2]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior

kable(autocorr.diag(mcmcHN2[[3]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior

kable(autocorr.diag(mcmcHN2[[4]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("n", "lambda 1", "lambda 2"),
  caption = "Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior

### Half-Normal (Informative: s21 = 10, s22 = 1)
set.seed(5, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
resultHN3 <- foreach(t = 1:4) %dopar% {

  start_time <- Sys.time()
  result <- gibbsHalfN(iter = 50000, s2_1 = 10, s2_2 = 1, dat = dat$disasters)
  run_time <- difftime(Sys.time(), start_time, units = "secs")

  list(run_time = run_time, result = result)

}
stopImplicitCluster()

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){resultHN3[[x]]$result[, 1]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("n"), titleLab = TeX("Trace plot: n with the informative prior"))

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){resultHN3[[x]]$result[, 2]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\\lambda_1$"), titleLab = TeX("Trace plot: $\\lambda_1$ with the informative prior"))

### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){resultHN3[[x]]$result[, 3]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\\lambda_2$"), titleLab = TeX("Trace plot: $\\lambda_2$ with the informative prior"))

```

```

p1
p2
p3

mcmcHN3 <- mcmc.list(lapply(1:4, function(x){mcmc(resultHN3[[x]]$result, start = 10001)}))
gelman.diag(mcmcHN3)

kable(autocorr.diag(mcmcHN3[[1]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 1 of the MCMC when using Half-Normal distribution as a prior")

kable(autocorr.diag(mcmcHN3[[2]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 2 of the MCMC when using Half-Normal distribution as a prior")

kable(autocorr.diag(mcmcHN3[[3]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 3 of the MCMC when using Half-Normal distribution as a prior")

kable(autocorr.diag(mcmcHN3[[4]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("n", "lambda 1", "lambda 2"),
      caption = "Autocorrelation of Chain 4 of the MCMC when using Half-Normal distribution as a prior")

### Plots
d1 <- sapply(1:4, function(x){resultHN1[[x]]$result[-c(1:10000), 1]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  group_by(Chain, Estimate) %>%
  summarise(n = n()) %>%
  ggplot(aes(x = Estimate, y = n)) +
  geom_bar(stat = "identity") +
  facet_grid(Chain ~ .) +
  labs(title = "Distribution of n", y = "Frequency", x = "n")

dPlot <- function(allChain, xLab, titleLab){
  ggplot(allChain, aes(x = Estimate, color = Chain)) +
    geom_density() +
    theme_bw() +
    theme(legend.position = "bottom") +
    labs(x = xLab, y = "Density", title = titleLab)
}

d2 <- sapply(1:4, function(x){resultHN1[[x]]$result[-c(1:10000), 2]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%
  dPlot(xLab = TeX("\$\lambda_1\$"), titleLab = TeX("Density plot: \$\lambda_1\$"))

d3 <- sapply(1:4, function(x){resultHN1[[x]]$result[-c(1:10000), 3]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  pivot_longer(paste0("Chain ", 1:4), names_to = "Chain", values_to = "Estimate") %>%

```

```

dPlot(xLab = TeX("$\\lambda_{[2]}$"), titleLab = TeX("Density plot: $\\lambda_{[2]}$"))

d1
d2
d3
sapply(1:4, function(x){apply(resultHN1[[x]]$result[-c(1:10000), ], 2, meanHDI)}) %>%
  `rownames<-`c("n", "ld 1", "ld 2")) %>%
  kable(col.names = paste0("Chain ", 1:4))

### Nimble: First Model
NB_mod1 <- nimbleCode({

  ### Prior
  theta ~ dunif(1, 112)
  alpha ~ dgamma(10, 10)
  lambda1 ~ dgamma(3, alpha)
  lambda2 ~ dgamma(3, alpha)

  ### Likelihood
  for (i in 1:N){
    X[i] ~ dpois(lambda1 * step(theta - i) + lambda2 * (1 - step(theta - i)))
  }

})

NB_const <- list(N = dim(dat)[1]) ### Constant
NB_data <- list(X = dat$disasters) ### Data
NB_init_mod1 <- list(theta = 20, alpha = 0.1, lambda1 = 0.1, lambda2 = 0.1)
save_param <- c("theta", "lambda1", "lambda2", "alpha")

### Run the model
set.seed(1)
start_time <- Sys.time()
NB_mod1_result <- nimbleMCMC(code = NB_mod1, constants = NB_const, data = NB_data,
                               inits = NB_init_mod1, niter = 50000, nburnin = 0,
                               nchains = 4, monitors = save_param)
NB_mod1_time <- difftime(Sys.time(), start_time, units = "secs")

NB_mod1_result[[1]][, 4] <- floor(NB_mod1_result[[1]][, 4])
NB_mod1_result[[2]][, 4] <- floor(NB_mod1_result[[2]][, 4])
NB_mod1_result[[3]][, 4] <- floor(NB_mod1_result[[3]][, 4])
NB_mod1_result[[4]][, 4] <- floor(NB_mod1_result[[4]][, 4])

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){floor(NB_mod1_result[[x]][, 4]))}) %>%
  `colnames<-`paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("n"), titleLab = TeX("Trace plot: n (Gamma Prior with Nimble)"))

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){NB_mod1_result[[x]][, 2]}) %>%

```

```

`colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\lambda_1$"), titleLab = TeX("Trace plot: $\\lambda_1$ (G

#### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){NB_mod1_result[[x]][, 3]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\lambda_2$"), titleLab = TeX("Trace plot: $\\lambda_2$ (G

#### Trace plot (all chains) for alpha
p4 <- sapply(1:4, function(x){NB_mod1_result[[x]][, 1]}) %>%
  `colnames<-`(paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ttGGPlot(burn_in = 10000, yLab = TeX("$\\alpha$"), titleLab = TeX("Trace plot: $\\alpha$ (Gamma Prior

grid.arrange(p1, p2, p3, p4)

mcmcNB1 <- mcmc.list(lapply(1:4, function(x){mcmc(NB_mod1_result[[x]], start = 10001)}))
gelman.diag(mcmcNB1)

kable(autocorr.diag(mcmcNB1[[1]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("alpha", "lambda 1", "lambda 2", "n"),
  caption = "Autocorrelation of Chain 1 of the MCMC when using Gamma prior with Nimble")

kable(autocorr.diag(mcmcNB1[[2]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("alpha", "lambda 1", "lambda 2", "n"),
  caption = "Autocorrelation of Chain 2 of the MCMC when using Gamma prior with Nimble")

kable(autocorr.diag(mcmcNB1[[3]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("alpha", "lambda 1", "lambda 2", "n"),
  caption = "Autocorrelation of Chain 3 of the MCMC when using Gamma prior with Nimble")

kable(autocorr.diag(mcmcNB1[[4]]), lags = c(0, 1, 5, 10, 100, 250)),
  col.names = c("alpha", "lambda 1", "lambda 2", "n"),
  caption = "Autocorrelation of Chain 4 of the MCMC when using Gamma prior with Nimble")
sapply(1:4, function(x){apply(NB_mod1_result[[x]][-c(1:10000), c(4, 2, 3, 1)], 2, meanHDI)}) %>%
  `rownames<-`(c("n", "ld 1", "ld 2", "alpha")) %>%
  kable(col.names = paste0("Chain ", 1:4))

### Nimble: Second Model
NB_mod2 <- nimbleCode({


  #### Prior
  theta ~ dunif(1, 112)
  lambda1 ~ T(dnorm(0, var = 1000), 0, )
  lambda2 ~ T(dnorm(0, var = 1000), 0, )
}

```

```

### Likelihood
for (i in 1:N){
  X[i] ~ dpois(lambda1 * step(theta - i) + lambda2 * (1 - step(theta - i)))
}

NB_const <- list(N = dim(dat)[1]) ### Constant
NB_data <- list(X = dat$disasters) ### Data
NB_init_mod2 <- list(theta = 20, lambda1 = 0.1, lambda2 = 0.1)
save_param <- c("theta", "lambda1", "lambda2")

### Run the model
set.seed(1)
start_time <- Sys.time()
NB_mod2_result <- nimbleMCMC(code = NB_mod2, constants = NB_const, data = NB_data,
                               inits = NB_init_mod2, niter = 50000, nburnin = 0,
                               nchains = 4, monitors = save_param)
NB_mod2_time <- difftime(Sys.time(), start_time, units = "secs")

NB_mod2_result[[1]][, 3] <- floor(NB_mod2_result[[1]][, 3])
NB_mod2_result[[2]][, 3] <- floor(NB_mod2_result[[2]][, 3])
NB_mod2_result[[3]][, 3] <- floor(NB_mod2_result[[3]][, 3])
NB_mod2_result[[4]][, 3] <- floor(NB_mod2_result[[4]][, 3])

### Trace plot (all chains) for n
p1 <- sapply(1:4, function(x){floor(NB_mod2_result[[x]][, 3])}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("n"), titleLab = TeX("Trace plot: n (HN Prior with Nimble)"))

### Trace plot (all chains) for lambda_1
p2 <- sapply(1:4, function(x){NB_mod2_result[[x]][, 1]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\\lambda_1$"), titleLab = TeX("Trace plot: $\\lambda_1$ (HN Prior with Nimble)"))

### Trace plot (all chains) for lambda_2
p3 <- sapply(1:4, function(x){NB_mod2_result[[x]][, 2]}) %>%
  `colnames<-` (paste0("Chain ", 1:4)) %>%
  as.data.frame() %>%
  mutate(iter = 1:50000) %>%
  pivot_longer(!iter, names_to = "Chain", values_to = "Estimate") %>%
  ggplot(burn_in = 10000, yLab = TeX("$\\lambda_2$"), titleLab = TeX("Trace plot: $\\lambda_2$ (HN Prior with Nimble)"))

p1
p2
p3

```

```

mcmcNB2 <- mcmc.list(lapply(1:4, function(x){mcmc(NB_mod2_result[[x]], start = 10001)}))
gelman.diag(mcmcNB2)

kable(autocorr.diag(mcmcNB2[[1]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("lambda 1", "lambda 2", "n"),
      caption = "Autocorrelation of Chain 1 of the MCMC when using HN prior with Nimble")

kable(autocorr.diag(mcmcNB2[[2]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("lambda 1", "lambda 2", "n"),
      caption = "Autocorrelation of Chain 2 of the MCMC when using HN prior with Nimble")

kable(autocorr.diag(mcmcNB2[[3]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("lambda 1", "lambda 2", "n"),
      caption = "Autocorrelation of Chain 3 of the MCMC when using HN prior with Nimble")

kable(autocorr.diag(mcmcNB2[[4]], lags = c(0, 1, 5, 10, 100, 250)),
      col.names = c("lambda 1", "lambda 2", "n"),
      caption = "Autocorrelation of Chain 4 of the MCMC when using HN prior with Nimble")
sapply(1:4, function(x){apply(NB_mod2_result[[x]][-c(1:10000), c(3, 1, 2)], 2, meanHDI)}) %>%
  `rownames<-`(`c("n", "ld 1", "ld 2")` %>%
  kable(col.names = paste0("Chain ", 1:4))

```