Explore Adaptive Models for Highway Traffic Volume via Bayesian Approach

Dingrui Tao

1 Introduction

A traffic system is complex, high-dimensional, and dynamic due to unpredictable factors involved in the system, such as weather and human interaction (Avila & Mezić, 2020). The highway system is a representative example. The main goal of a highway system is to optimize traffic efficiency and reduce congestion. Under this objective, traffic flow (also referred to as traffic volume) prediction is a crucial part of improving the system. There exists a key challenge for traffic flow prediction networks, which is how to construct an adaptive model based on historical data. (Kong et al., 2019). In this project, the aim is to explore such an adaptive model via the Bayesian approach.

2 Dataset

The dataset records the hourly traffic volume for the westbound I-94 highway in Minneapolis-St Paul, MN, including weather and holiday features. This dataset is downloaded from the UCI Machine Learning Repository.[1]

There are in total 9 features and 48204 rows of observations in the dataset. The detail of features is presented in Table 1. There are no missing values in the data. But the numeric summary shows that there are some abnormal observations in the data: some observations' temperature is exactly 0K, which is physically impossible; There is another observation with an hourly rain amount of 9831.3 mm, which is nearly 30 times the current world record for hourly rain peak amount.

Firstly, I removed observations with abnormal temperature and hourly rain amount. Then I do some visualizations between variables in the dataset. The first plot is a histogram of the response variable hourly traffic volume in the data, shown in Figure 2; The second plot is a boxplot of hourly traffic volume in different holidays, shown in Figure 3; The third plot is another boxplot of hourly traffic under different weather, shown in Figure 4

[1] `https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume`

3 Methods

This study consists of two parts. The first part is Ordinary Bayesian inference. I directly estimate the posterior distribution of the average hourly traffic volume by ordinary Bayesian Inference; The second part is hierarchical Bayesian inference. I assume there is a hierarchical structure in different holidays/ under different weather, then use a hierarchical Bayesian model to estimate the posterior distribution.

3.1 Ordinary Bayesian Inference

In this part, I establish an ordinary Bayesian model that uses prior information and the data to infer the target average hourly traffic volume λ. The model is formulated as:

$$f(\lambda|y_1, ..., y_n) \propto f(\lambda)f(y_1, ..., y_n|\lambda) \tag{1}$$

For the prior distribution $f(\lambda)$, since there is not plenty of information about traffic flow in partial highway in the fixed time period, I used Gamma prior with randomly initialized parameters. In this context, the average parameter λ only makes sense if $\lambda > 0$. Therefore, the prior distribution with support $(0, +\infty)$ is formulated as:

$$\lambda \sim Gamma(2, 1) \tag{2}$$

For the likelihood function, my dataset records discrete traffic volume in a fixed time interval, so Poisson likelihood may be suitable here, which is formulated as:

$$y_i|\lambda \overset{iid}{\sim} Poisson(\lambda),\ i = 1, ..., n \tag{3}$$

Implementing the algorithm in R via Stan file, then I perform a posterior predictive check to assess the prediction performance.

3.2 Hierarchical Bayesian Inference

Based on the boxplot of traffic volume under different weather conditions, I believe there may exist a hierarchical structure in the data, while the ordinary Bayesian inference cannot detect it. The structure is shown in Figure 1.

Implementing the algorithm in R via Stan file, then I perform the posterior predictive check for each weather type, as well as comparison with the ordinary Bayesian model.

4 Results

4.1 Ordinary Bayesian Inference

The algorithm converges with 4 chains and 7000 iterations per chain as shown in the Trace plot Figure 5. The effect size for λ is respectively 4687.297.

For efficiency, the total runtime for the warmup and sampling phase is respectively 72.487 seconds and 76.895 seconds. This algorithm is quite efficient and has space to be improved because current iterations are not the minimum it needs to converge.

For evaluation, as shown in the Histogram 6 the posterior predictive distribution captures the true average of traffic volume in the dataset very well: the true mean nearly falls at the center of the posterior distribution.

4.2 Hierarchical Bayesian Inference

The algorithm converges with 4 chains and 1000 number of iterations per chain as shown in the Trace plot Figure 7. The effect size for λ_i is respectively 816.9327, 1798.1752, 1931.5591, 2319.3443, 2273.0209, 2540.4069, 2278.4052, 2329.9722, 2541.9449, 2207.7909, 1315.9222, 610.4568, for $i \in [0, 11]$.

For efficiency, the total runtime for warmup and sampling phase is respectively 488.113 seconds and 472.757 seconds. This algorithm is not very efficient and does take some time to converge.

For evaluation, from Figure 8, the predictive performance of the ordinary model is not ideal: the posterior samples only capture the true average under "Smoke" weather, missing the true average under other 10 weather types; From Figure 9, the predictive performance of the hierarchical model is quite satisfactory: average traffic volume under each weather type was captured by the predictive distribution.

5 Conclusion/Limitation/Future work

In general, considering overall performance and efficiency, the ordinary Bayesian model could be a good adaptive model for traffic flow prediction. However, if there is a need for specific average traffic under different weather, which is usually more practical in reality, the hierarchical model is a better choice.

There are several limitations to this project. The first one is the time variable that appears in the dataset while this project does not utilize it to explore any potential time series relationship; The second one is the prior choice. The Bayesian model's prior distribution depends on prior information. Without enough information, the prior may be inappropriate and thus bias the result; The third one is the computation efficiency. Compared with common statistical methods such as regression, the compile speed for Bayesian methods is too slow.

Future works can be contributed to many aspects. Firstly, the prior choice needs to be improved. Secondly, this project only explores the hierarchical structure related to weather while there may exist such structure for another variable such as Holidays. Thirdly, this project mainly answers the inference problem, future work can be focused on prediction via Bayesian approaches such as Bayesian Regression, etc.

References

Avila, A.M., Mezić, I. Data-driven analysis and forecasting of highway traffic dynamics. Nat Commun 11, 2090 (2020). https://doi.org/10.1038/s41467-020-15582-5

Kong, F., Li, J., Jiang, B., Zhang, T., & Song, H. (2019). Big data-driven machine learning-enabled traffic flow prediction. Transactions on Emerging Telecommunications Technologies, 30(9), e3482. https://doi.org/10.1002/ett.3482

A Table & Figure

Feature Name	Data Type	Description
holiday	Categorical	US holidays[2]
temp	Continuous	average temperature[3]
rain_1h	Continuous	hourly rain amount[4]
snow_1h	Continuous	hourly snow amount[5]
clouds_all	Integer	cloud cover rate[6]
weather_main	Categorical	general weather type[7]
weather_description	Categorical	specific weather type[8]
date_time	Date	local CST time[9]
traffic_volume	Integer	traffic volume[10]

Table 1: Feature Description

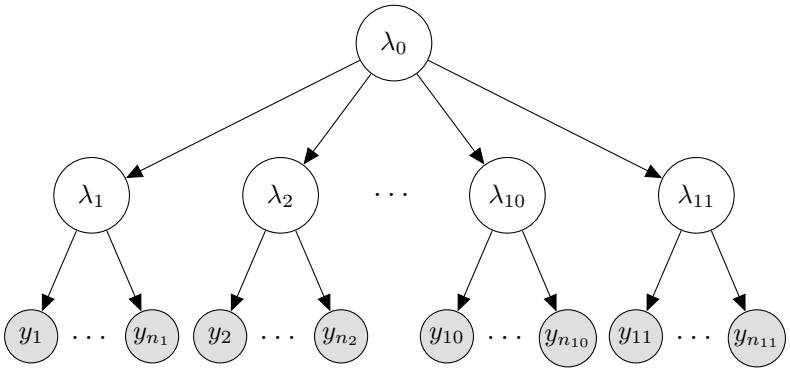


Figure 1: Structure of the hierarchical model.

[2]US National holidays plus regional holiday, Minnesota State Fair
[3]in Kelvin
[4]in mm
[5]in mm
[6]in x100% percentage
[7]Short general textual description of the current weather
[8]long specific textual description of the current weather
[9]recorded in each hour
[10]Hourly I-94 ATR 301 reported westbound traffic volume

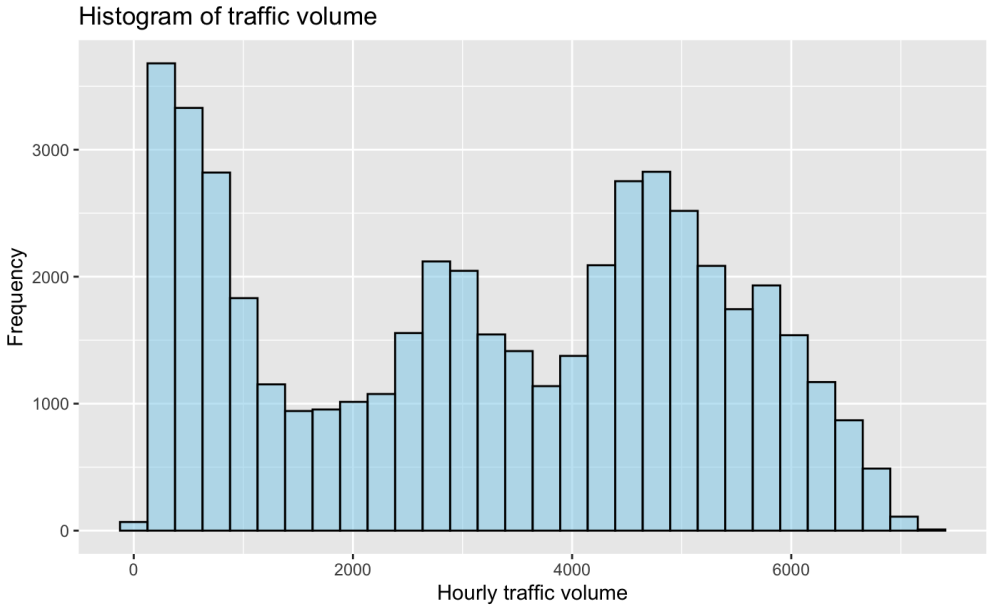


Figure 2: Histogram of Hourly Traffic Volume

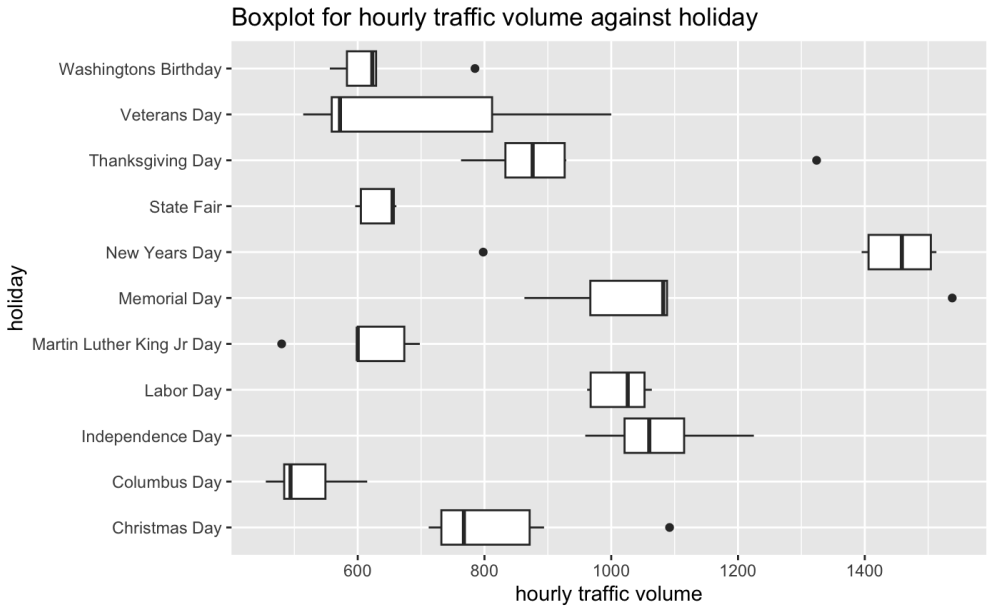


Figure 3: Boxplot of Hourly Traffic Volume against Holiday

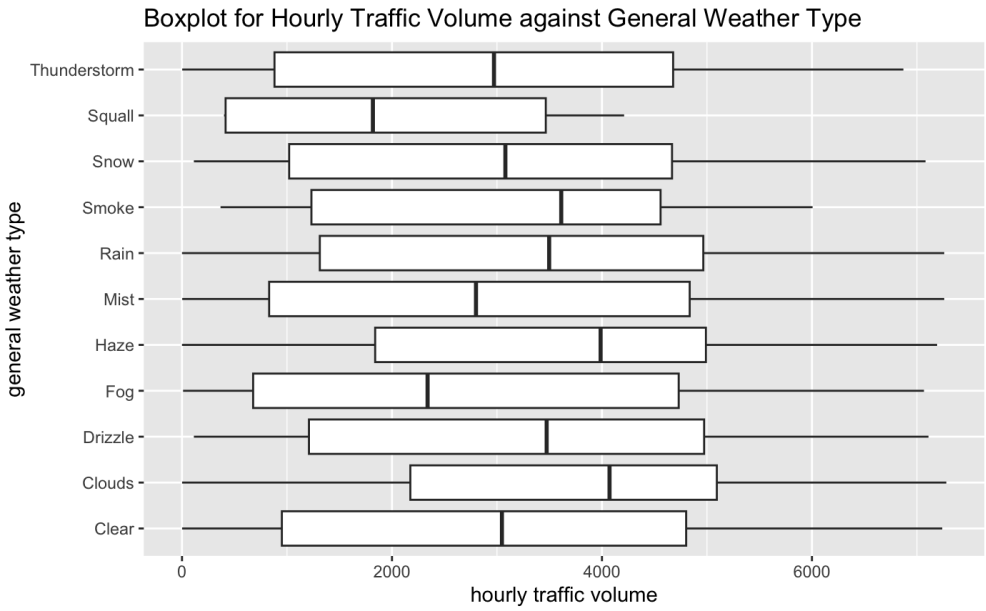


Figure 4: Boxplot of Hourly Traffic Volume against Weather

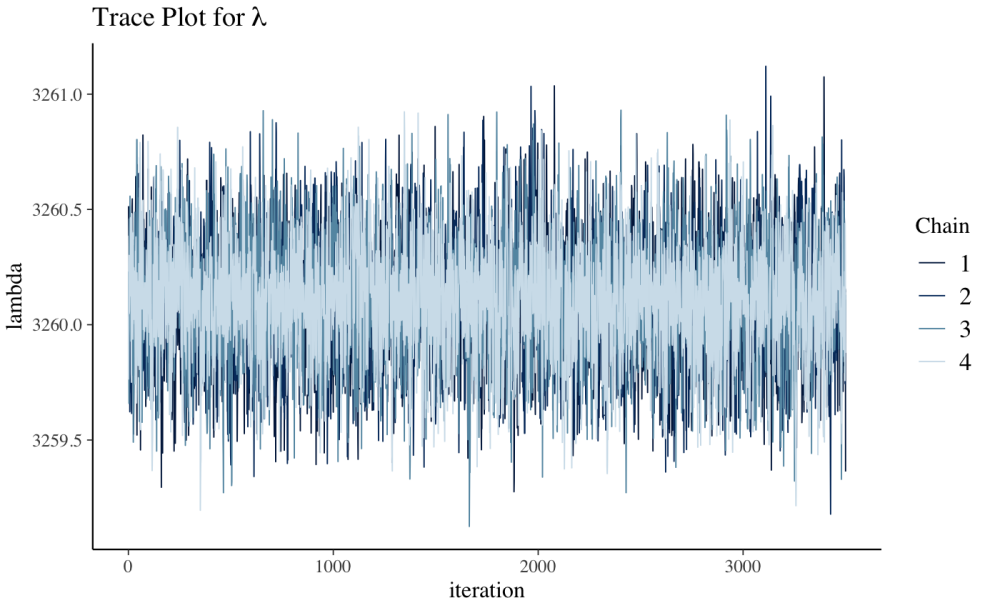


Figure 5: Trace plot for μ and σ

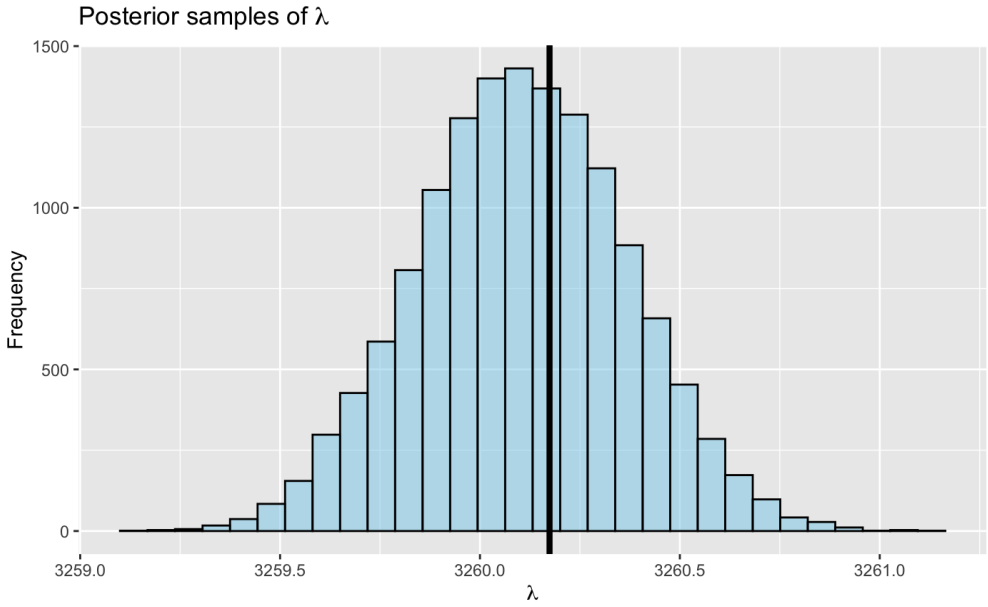


Figure 6: Histogram of posterior distributions with true average of λ

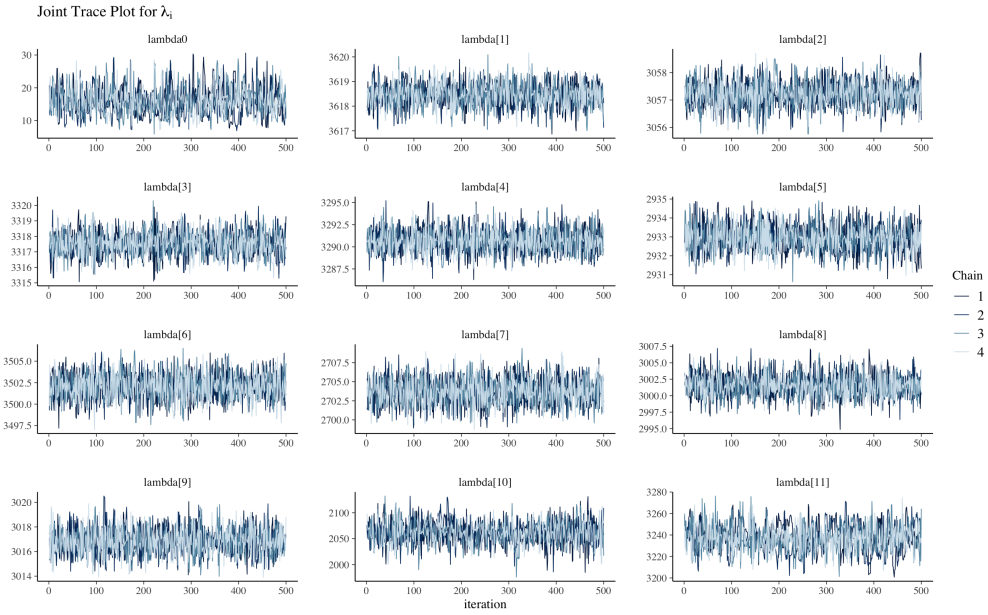


Figure 7: Trace plot of λ_i

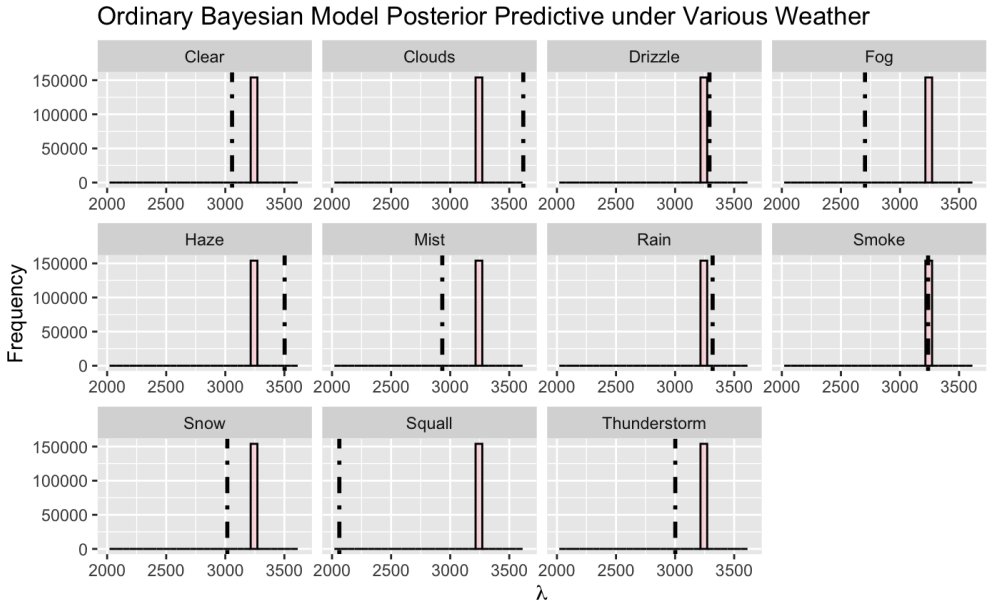


Figure 8: Histogram of posterior predictive for ordinary model

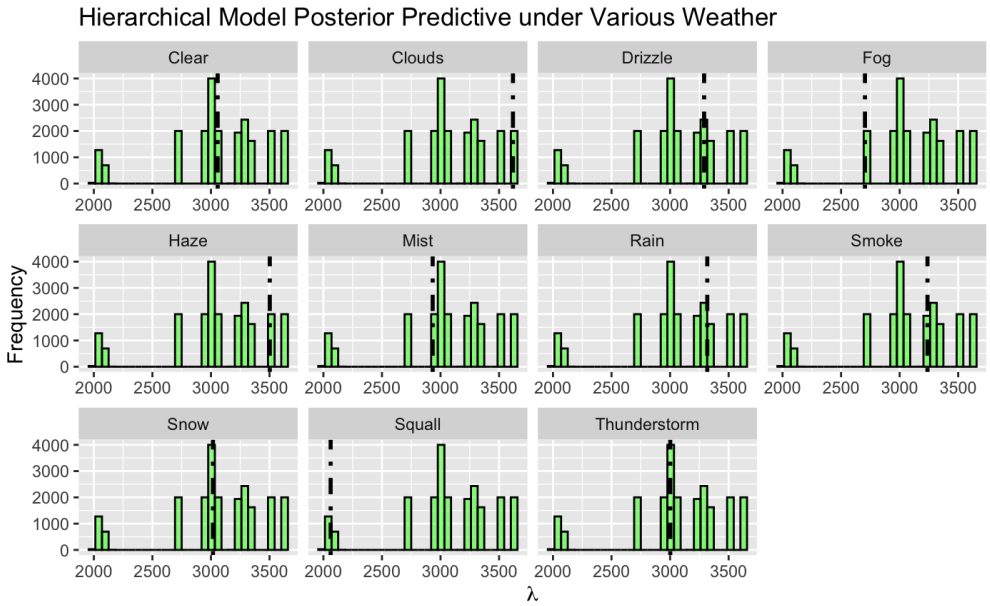


Figure 9: Histogram of posterior predictive for hierarchical model

B R Code

2 Dataset

```
# read the data
Highway_data <- read.csv("Metro_Interstate_Traffic_Volume.csv",
                         header = T)
head(Highway_data)
```

```
##   holiday   temp rain_1h snow_1h clouds_all weather_main weather_description
## 1    None 288.28       0       0         40       Clouds    scattered clouds
## 2    None 289.36       0       0         75       Clouds       broken clouds
## 3    None 289.58       0       0         90       Clouds     overcast clouds
## 4    None 290.13       0       0         90       Clouds     overcast clouds
## 5    None 291.14       0       0         75       Clouds       broken clouds
## 6    None 291.72       0       0          1        Clear        sky is clear
##             date_time traffic_volume
## 1 2012-10-02 09:00:00           5545
## 2 2012-10-02 10:00:00           4516
## 3 2012-10-02 11:00:00           4767
## 4 2012-10-02 12:00:00           5026
## 5 2012-10-02 13:00:00           4918
## 6 2012-10-02 14:00:00           5181
```

```
# check dimension of the data
dim(Highway_data)
```

```
## [1] 48204     9
```

```
# number of missing values in each column
kable(tibble("column" = colnames(Highway_data),
             "# of missing values" =
               c(sum(is.na(Highway_data$holiday)),
                 sum(is.na(Highway_data$temp)),
                 sum(is.na(Highway_data$rain_1h)),
                 sum(is.na(Highway_data$snow_1h)),
                 sum(is.na(Highway_data$clouds_all)),
                 sum(is.na(Highway_data$weather_main)),
                 sum(is.na(Highway_data$weather_description)),
                 sum(is.na(Highway_data$date_time)),
                 sum(is.na(Highway_data$traffic_volume)))))
```

column	# of missing values
holiday	0
temp	0
rain_1h	0
snow_1h	0
clouds_all	0
weather_main	0
weather_description	0
date_time	0

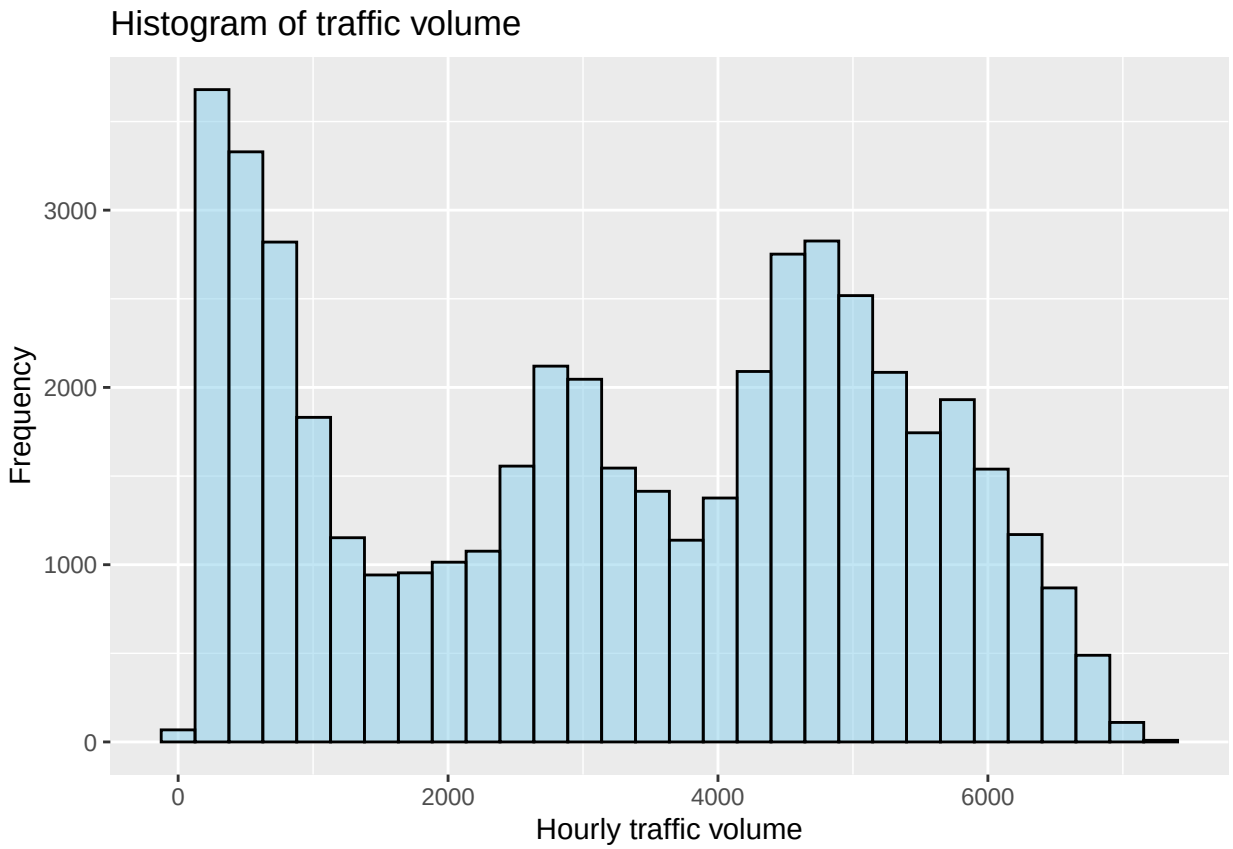
column	# of missing values
traffic_volume	0

```r
# An overview of each column of data
summary(Highway_data)
```

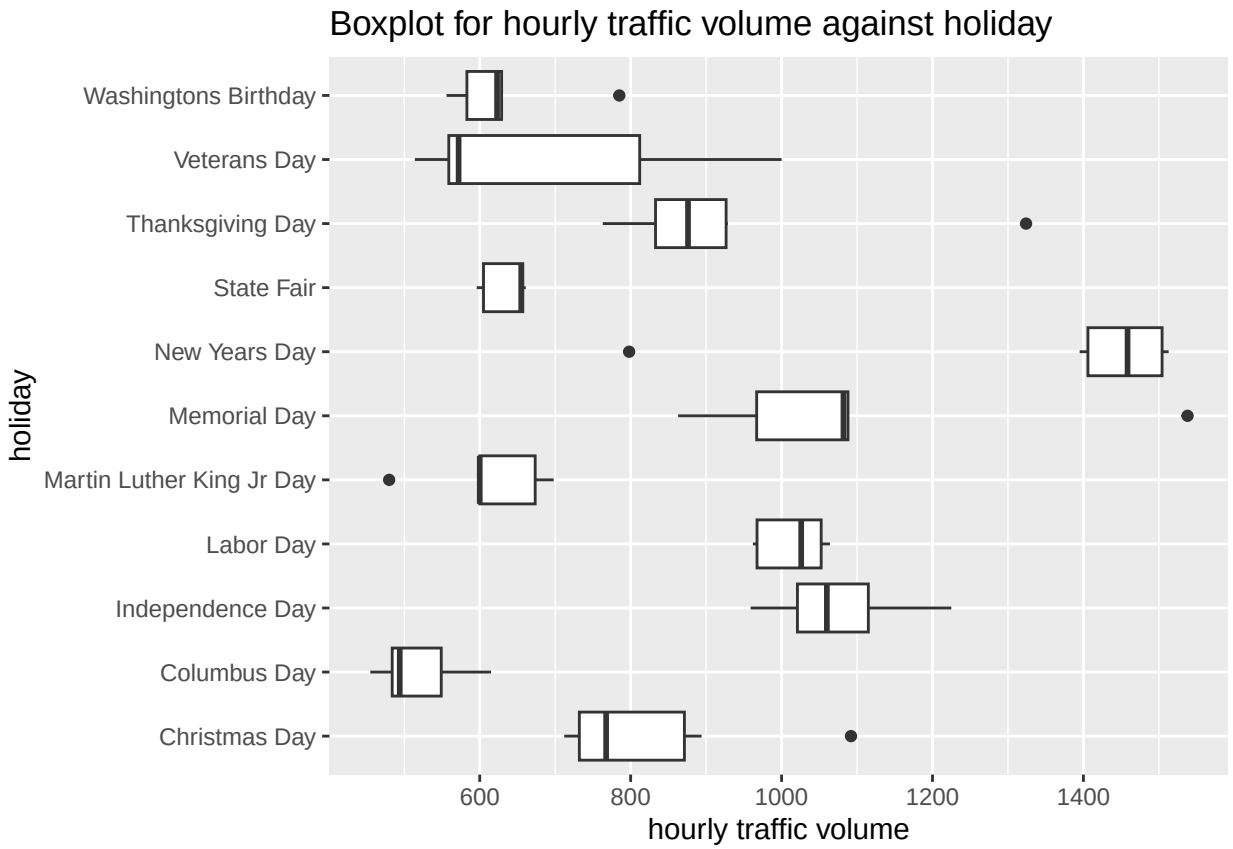
```
##    holiday               temp          rain_1h            snow_1h
##  Length:48204       Min.   :  0.0   Min.   :   0.000   Min.   :0.0000000
##  Class :character   1st Qu.:272.2   1st Qu.:   0.000   1st Qu.:0.0000000
##  Mode  :character   Median :282.4   Median :   0.000   Median :0.0000000
##                     Mean   :281.2   Mean   :   0.334   Mean   :0.0002224
##                     3rd Qu.:291.8   3rd Qu.:   0.000   3rd Qu.:0.0000000
##                     Max.   :310.1   Max.   :9831.300   Max.   :0.5100000
##    clouds_all     weather_main       weather_description  date_time
##  Min.   :  0.00   Length:48204       Length:48204        Length:48204
##  1st Qu.:  1.00   Class :character   Class :character    Class :character
##  Median : 64.00   Mode  :character   Mode  :character    Mode  :character
##  Mean   : 49.36
##  3rd Qu.: 90.00
##  Max.   :100.00
##  traffic_volume
##  Min.   :   0
##  1st Qu.:1193
##  Median :3380
##  Mean   :3260
##  3rd Qu.:4933
##  Max.   :7280
```

```r
# Remove observations with abnormal temperature and hourly rain amount
highway_traffic <- Highway_data[Highway_data$temp > 0 & Highway_data$rain_1h < 200, ]
```

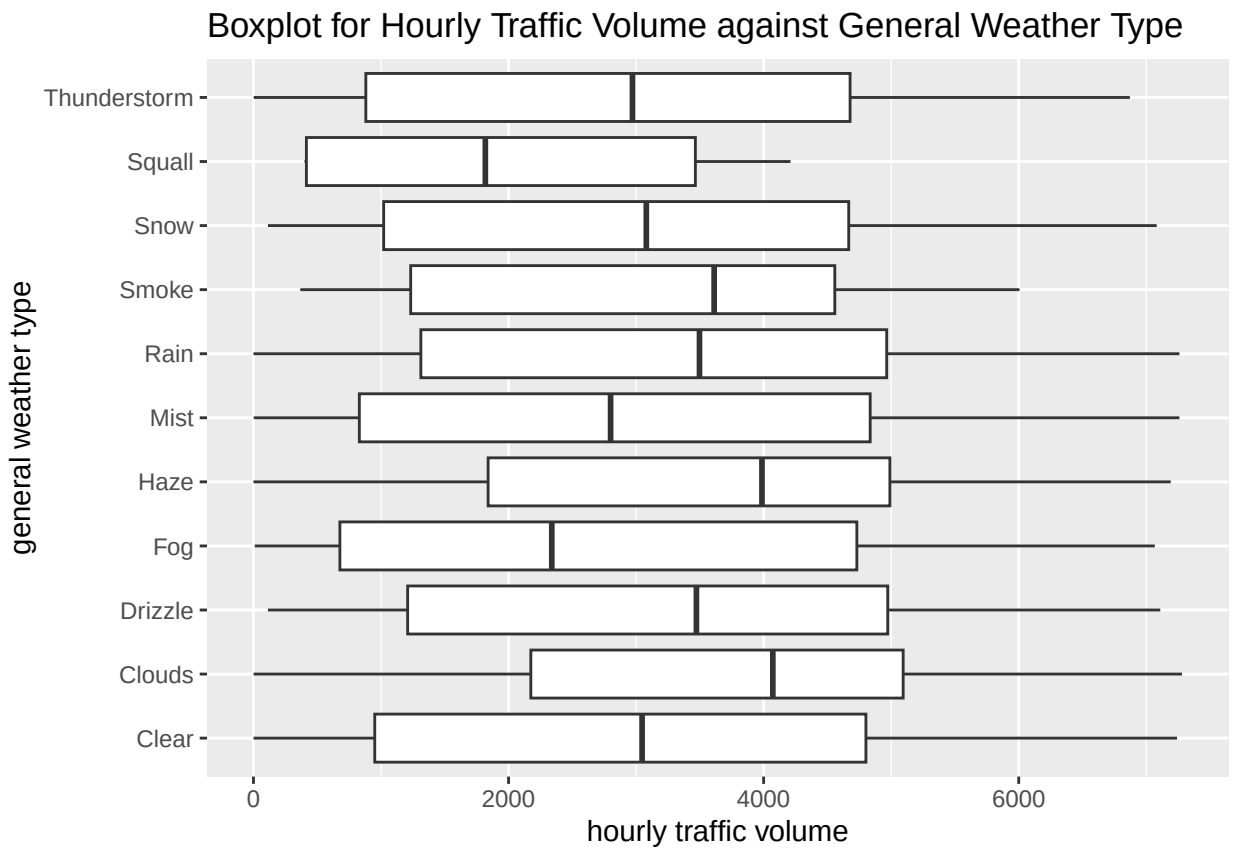
```r
# a few visualizations
# Visualize response variable traffic volume
ggplot(highway_traffic) +
  geom_histogram(aes(x = traffic_volume),
                 bins = 30, fill = "skyblue", color = "black", alpha = 0.5) +
  labs(title = "Histogram of traffic volume",
       x = "Hourly traffic volume", y = "Frequency")
```




```
# boxplot of traffic volume against holiday
b1 <- ggplot(data = highway_traffic[highway_traffic$holiday != "None", ]) +
  geom_boxplot(aes(x = holiday, y = traffic_volume)) +
  labs(x = "holiday", y = "hourly traffic volume") +
  ggtitle("Boxplot for hourly traffic volume against holiday") +
  coord_flip()
b1
```




```
# boxplot of traffic volume against general weather type
b2 <- ggplot(data = highway_traffic) +
  geom_boxplot(aes(x = weather_main, y = traffic_volume)) +
  labs(x = "general weather type", y = "hourly traffic volume") +
  ggtitle("Boxplot for Hourly Traffic Volume against General Weather Type") +
  coord_flip()
b2
```



3 Methods

3.1 Ordinary Bayesian Inference

```
# R Stan file for the model
poisson_stan <- "Poisson.stan"
writeLines(readLines(poisson_stan))
```

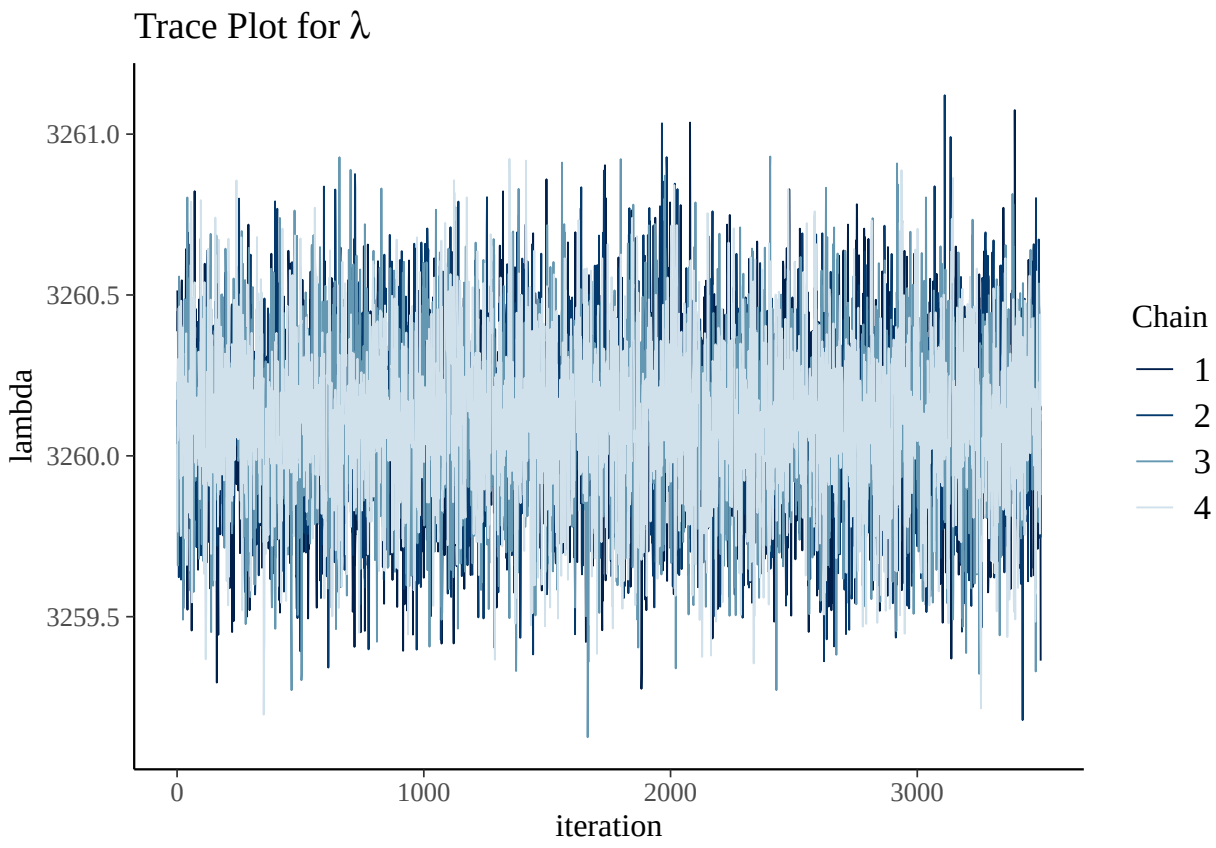
```
## data {
##   int<lower=0> N;  // number of observations
##   int<lower=0> y[N];  // data vector y
##
## }
## parameters {
##   real<lower = 0> lambda; // mean parameter
## }
## model {
##   // model
##   for (n in 1:N) {
##     y[n] ~ poisson(lambda);
##   }
##   // prior
##   lambda ~ gamma(2, 1);
```

```
##
## }
```

```
# fit the model with Stan
fit_poisson <- stan(file = poisson_stan,
                    data = list(N = nrow(highway_traffic),
                                y = highway_traffic$traffic_volume),
                  seed = SEED, chains = 4, iter = 7000)
```

```
## Trying to compile a simple C file
```

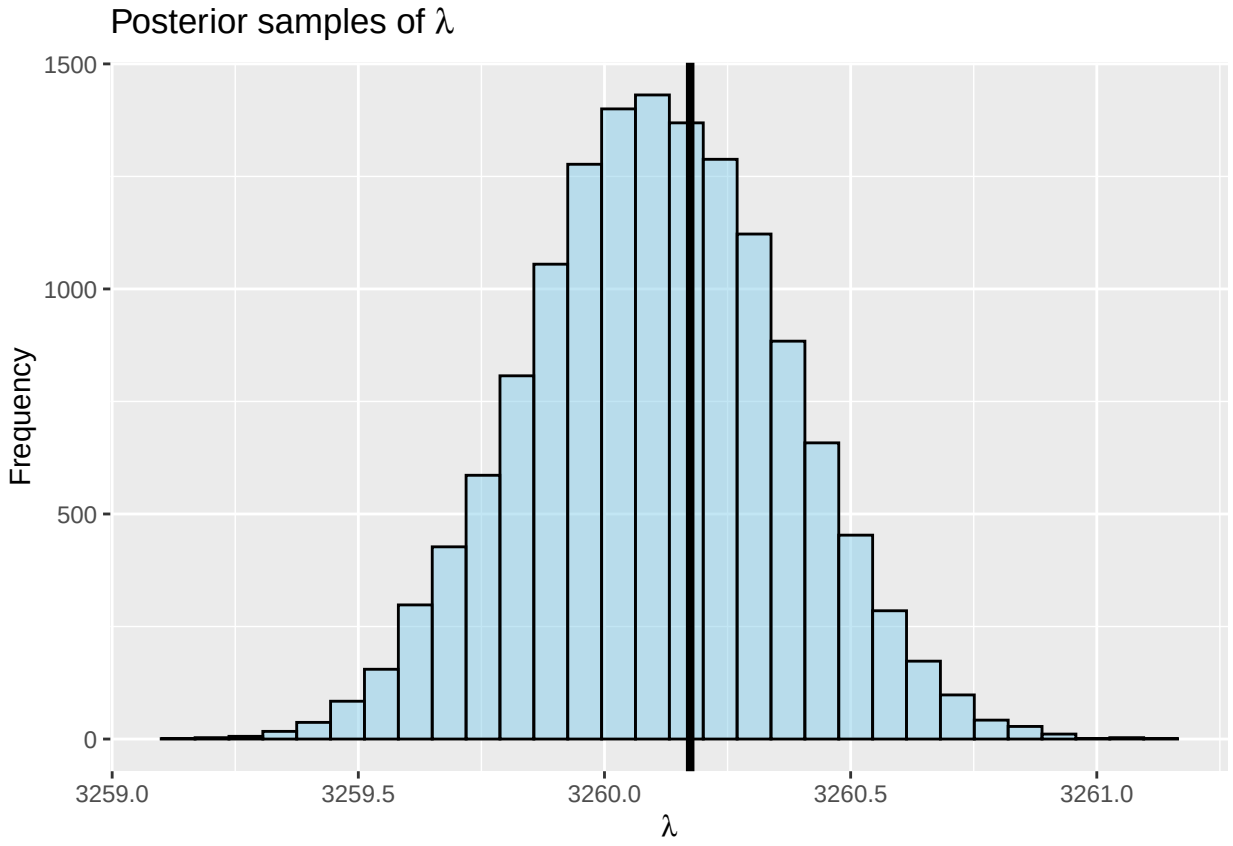
```
lambda_stan <- as.data.frame(fit_poisson, par = "lambda")
lambda_stan$iter <- seq(1, nrow(lambda_stan), 1)

# monitor convergence by trace plot
mcmc_trace(fit_poisson, par = c("lambda")) +
  labs(x = "iteration", title = bquote("Trace Plot for " * lambda))
```




```
# check effect size
summary(fit_poisson)$summary[, "n_eff"]
```

```
##   lambda     lp__
## 4687.297 6304.433
```

```r
# Posterior predictive check
ggplot(lambda_stan) +
  geom_histogram(aes(x = lambda),  bins = 30, fill = "skyblue", color = "black", alpha = 0.5) +
  labs(x=expression(lambda), y = "Frequency") +
  ggtitle(bquote("Posterior samples of " * lambda)) +
  geom_vline(xintercept = mean(highway_traffic$traffic_volume), linewidth = 1.5)
```




```r
# runtime check
ordinary_runtime <- get_elapsed_time(fit_poisson)
rbind(ordinary_runtime, colSums(ordinary_runtime))
```

```
##          warmup sample
## chain:1 10.912 11.182
## chain:2 10.406 10.672
## chain:3 10.476 10.608
## chain:4 10.521 11.943
##         42.315 44.405
```

3.2 Hierarchical Bayesian Inference

```r
# Create a new column with the index for each level in weather_main
highway_traffic$index <- as.numeric(factor(highway_traffic$weather_main,
                                           levels = unique(highway_traffic$weather_main)))
```

```r
# view the weather description and corresponding index
kable(data.frame(weather = unique(highway_traffic$weather_main),
                 index = seq(1, length(unique(highway_traffic$weather_main)), 1)))
```

weather	index
Clouds	1
Clear	2
Rain	3
Drizzle	4
Mist	5
Haze	6
Fog	7
Thunderstorm	8
Snow	9
Squall	10
Smoke	11

```r
# Implementing hierarchical model via Stan
hier <- "Hier_fit.stan"
writeLines(readLines(hier))
```

```
## data {
##   int<lower=0> N;                  // Number of observations
##   int<lower=0> J;                  // Number of EV models (groups)
##   int<lower=1, upper=J> weather_type[N]; // Category index for each observation
##   int<lower=0> volume[N];          // Observed count data
## }
##
## parameters {
##   real<lower=0> lambda0;              // Overall mean (rate parameter)
##   real<lower=0> tau;               // Between-group SD
##   real<lower=0> lambda[J];         // Group-specific rate parameters
## }
##
## model {
##   // Priors
##   lambda0 ~ gamma(5, 1);              // Prior for overall mean (Gamma prior)
##   tau ~ gamma(2, 1);               // Prior for between-group SD
##
##   // Hierarchical structure for group-specific rates
##   lambda ~ gamma(lambda0 * tau, tau); // Gamma prior for lambda[J]
##
##   // Poisson likelihood
##   for (n in 1:N) {
##     volume[n] ~ poisson(lambda[weather_type[n]]);
##   }
## }
```

```r
# fit the hierarchical model with Stan
fit_hier <- stan(file = hier,
       data = list(N = nrow(highway_traffic),
```

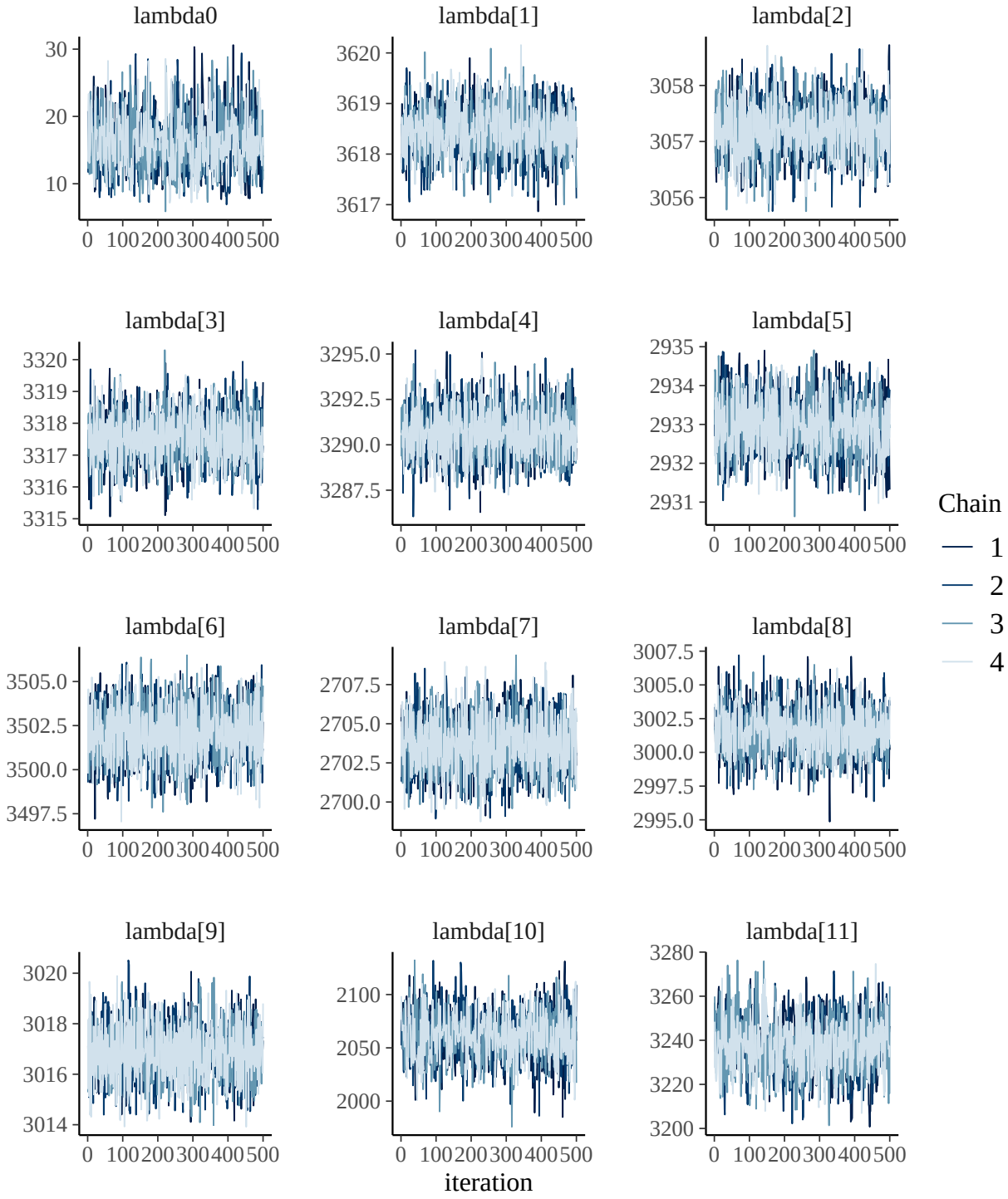
```
       J = length(levels(factor(highway_traffic$weather_main))),
       weather_type = highway_traffic$index,
       volume = highway_traffic$traffic_volume),
       seed = SEED, chains = 4, iter = 1000, control = list(max_treedepth = 15))
```

```
## Trying to compile a simple C file
```

```
# extract posterior samples
params <- c("lambda0", "lambda[1]", "lambda[2]", "lambda[3]", "lambda[4]", "lambda[5]", "lambda[6]",
            "lambda[7]", "lambda[8]", "lambda[9]", "lambda[10]", "lambda[11]")
lambda_hier <- as.data.frame(fit_hier, par = params)
lambda_hier$iter <- seq(1, nrow(lambda_hier), 1)
```

```
# monitor joint convergence
mcmc_trace(fit_hier, par = params, facet_args = list(nrow = 4, ncol = 3)) +
  labs(title = bquote("Joint Trace Plot for " * lambda[i]), x = "iteration")
```

Joint Trace Plot for λ_i




```
# check effect size
summary(fit_hier)$summary[, "n_eff"]
```

```
##    lambda0         tau   lambda[1]   lambda[2]   lambda[3]   lambda[4]   lambda[5]
##   786.1955  1146.7087   1773.4365   1888.6264   2161.9362   2436.4156   2254.1924
##  lambda[6]   lambda[7]   lambda[8]   lambda[9]  lambda[10]  lambda[11]        lp__
```

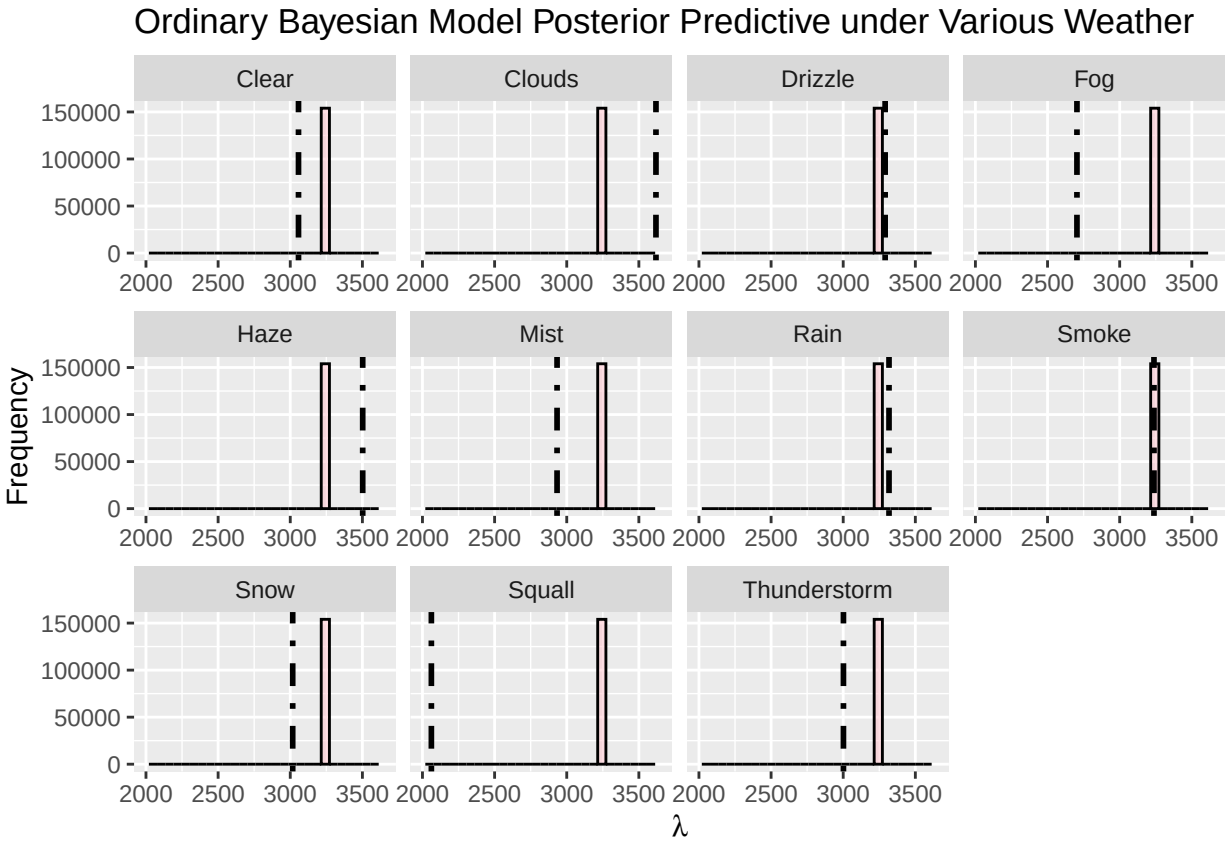
```
##  2426.1438  2154.5126  2230.4464  2436.0242  1255.7559   581.6392   725.2206
```

```r
# runtime check
hier_runtime <- get_elapsed_time(fit_hier)
rbind(hier_runtime, colSums(hier_runtime))
```

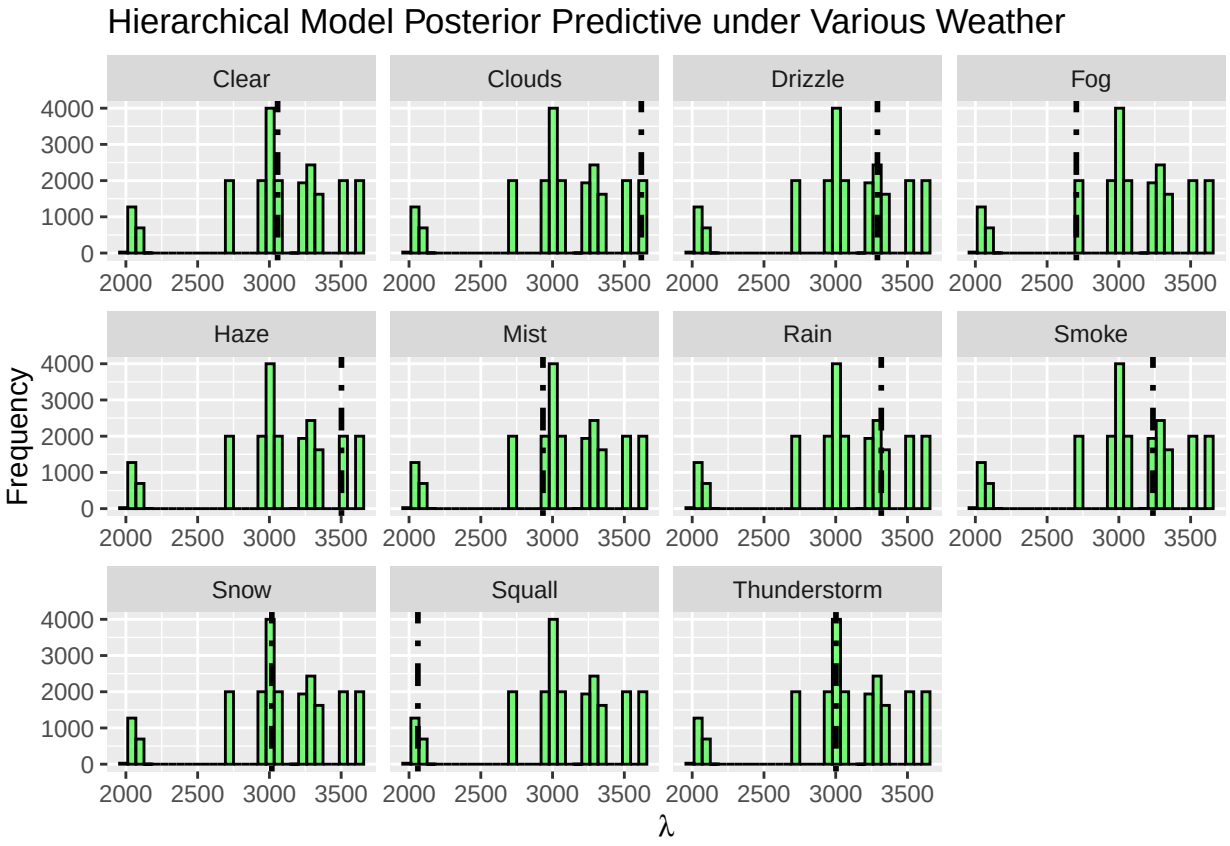
```
##          warmup  sample
## chain:1  34.386  99.973
## chain:2  83.493  21.012
## chain:3  43.498  43.154
## chain:4  51.413  43.496
##         212.790 207.635
```

```r
# Posterior predictive check under different weather type on ordinary model
plot_dat_lambda <- data.frame(volume = rep(lambda_stan$lambda, 11),
                      group = rep(seq(1, 11, 1), each = length(lambda_stan$lambda)))


test <- highway_traffic %>%
  group_by(weather_main) %>%
  summarize(
    vertical_line = mean(traffic_volume))

vline <- test[c(2, 1, 7, 3, 6, 5, 4, 11, 9, 10, 8), ]

# posterior predictive for one parameter model
suppressWarnings(ggplot(plot_dat_lambda) +
  # Histogram
  geom_histogram(
    aes(x = volume), bins = 30, fill = "pink", color = "black", alpha = 0.5) +
  geom_vline(data = vline, aes(xintercept = vertical_line),
             size = 1, linetype = "dotdash") +
  facet_wrap(~ weather_main, scales = "free_x") +
  labs(title =
         "Ordinary Bayesian Model Posterior Predictive under Various Weather",
       x = expression(lambda), y = "Frequency") +
  scale_x_continuous(limits = c(2000, 3650)))
```

```
## Warning: Removed 22 rows containing missing values (‘geom_bar()‘).
```




```
# Posterior predictive check under different weather type on hierarchical model
plot_dat_hier <- data.frame(volume = c(lambda_hier$`lambda[1]`,
                                       lambda_hier$`lambda[2]`,
                                       lambda_hier$`lambda[3]`,
                                       lambda_hier$`lambda[4]`,
                                       lambda_hier$`lambda[5]`,
                                       lambda_hier$`lambda[6]`,
                                       lambda_hier$`lambda[7]`,
                                       lambda_hier$`lambda[8]`,
                                       lambda_hier$`lambda[9]`,
                                       lambda_hier$`lambda[10]`,
                                       lambda_hier$`lambda[11]`),
                            group = rep(seq(1, 11, 1),
                                        each = length(lambda_hier$`lambda[1]`)))

# posterior predictive for hierarchical model
ggplot(plot_dat_hier) +
  # Histogram
  geom_histogram(
    aes(x = volume), bins = 30, fill = "green", color = "black", alpha = 0.5) +
  geom_vline(data = vline, aes(xintercept = vertical_line), size = 1, linetype = "dotdash") +
  facet_wrap(~ weather_main, scales = "free_x") +
  labs(title = "Hierarchical Model Posterior Predictive under Various Weather",
       x = expression(lambda), y = "Frequency")
```


Hierarchical Model Posterior Predictive under Various Weather
Clear
Clouds
Drizzle
Fog
Haze
Mist
Rain
Smoke
Snow
Squall
Thunderstorm
Frequency
λ
0
1000
2000
3000
4000
2000
2500
3000
3500