Bayesian models for predictions of Amazon Forest fires

Contents

1 Abstract

The aim of this project is to analyze the number of fires in the states of Brazil and build a Bayesian model in order to make predictions about the frequency of forest fires in a time series. This kind of prediction could help to take action to prevent them.

2 Introduction

The dataset on which the analysis are made is taken from "Kaggle", an online community of data scientists and machine learners where many data sets are available for the users. The "amazon" dataset reports the number of forest fires in Brazil divided by states. The series comprises the period of approximately 10 years (1998 to 2017). The data were obtained from the official website of the Brazilian government. Brazil has the largest rainforest on the planet that is the Amazon rainforest. Fires are a serious problem for the preservation of the Tropical Forests. Understanding the frequency of forest fires in a time series can help to take action to prevent them.

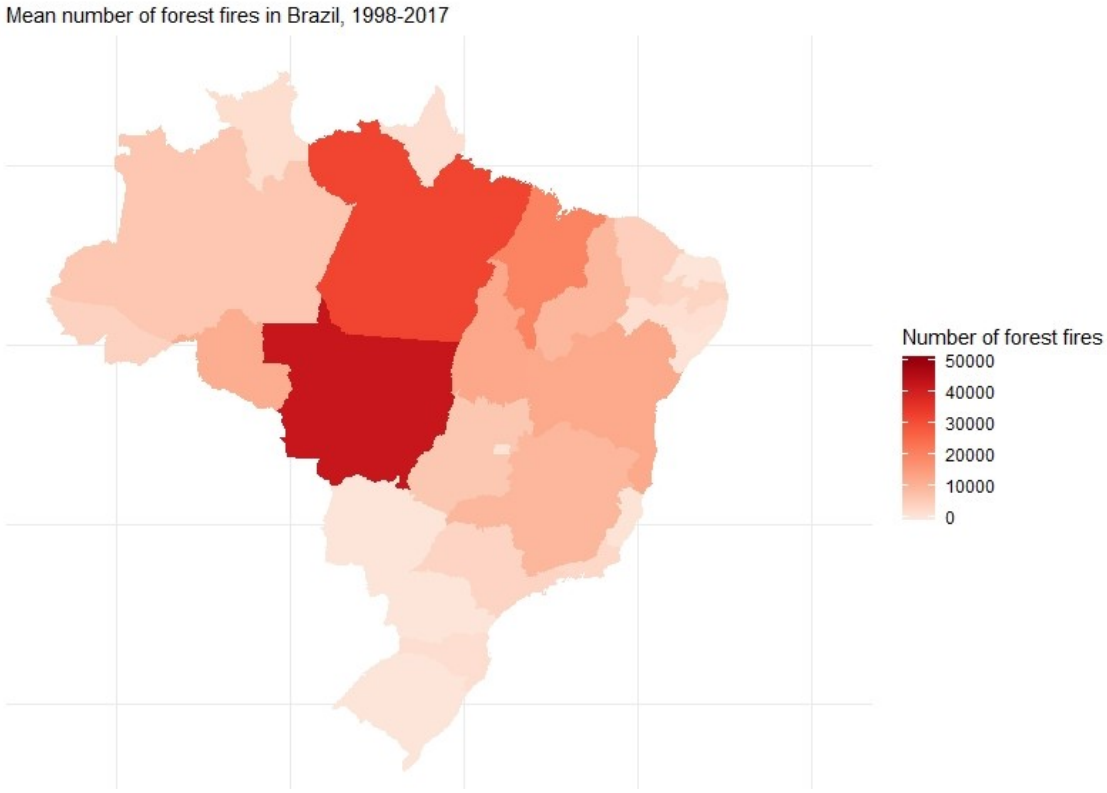
3 Explorative Analysis

In the original dataset "Amazon", the following features are present:

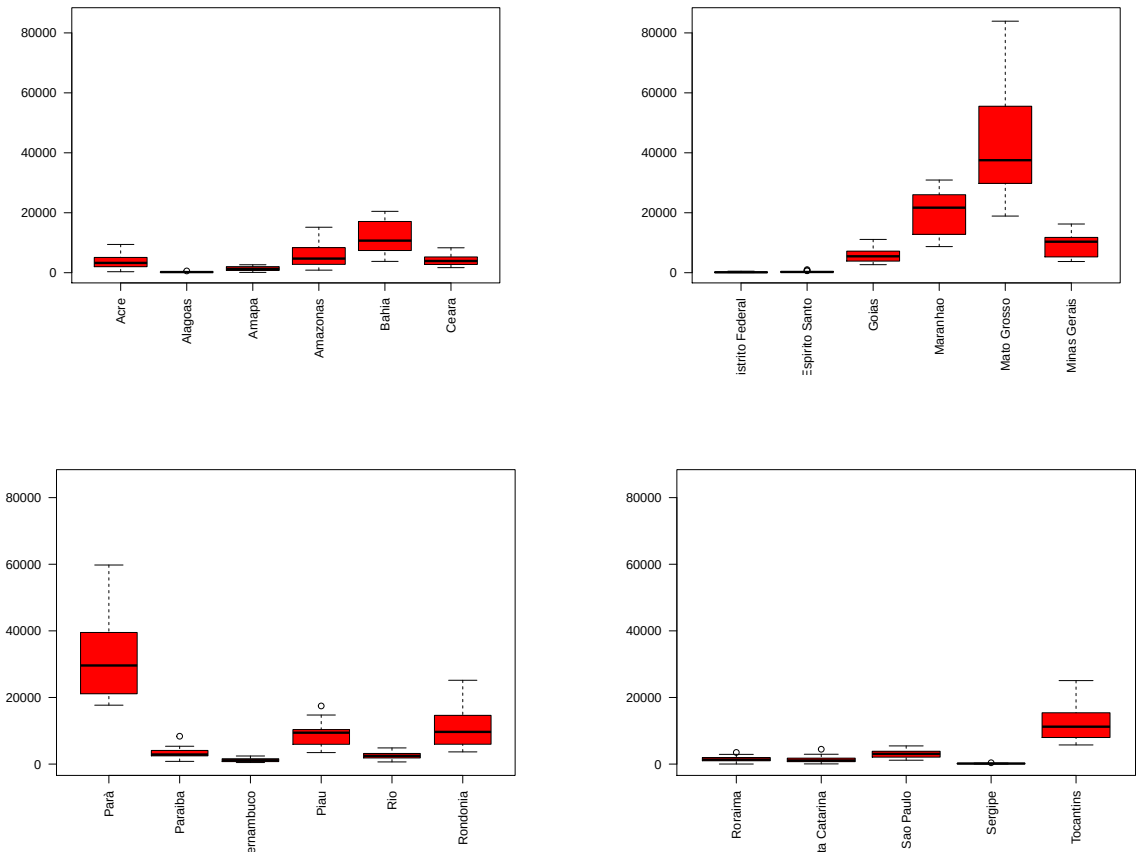
- year (1998 - 2017);
- state (23);
- month;
- number of fires;
- date;

All the following analysis are made on a subset of the initial dataset. For each state and for each year, the sum of the total fires per month is calculated. Moreover, in order to apply the Stan code, a new dataset has been calculated: each state is treated as a different group and for each state an index has been assigned. Separate, Pooled and Hierarchical models will be applied on this dataset.

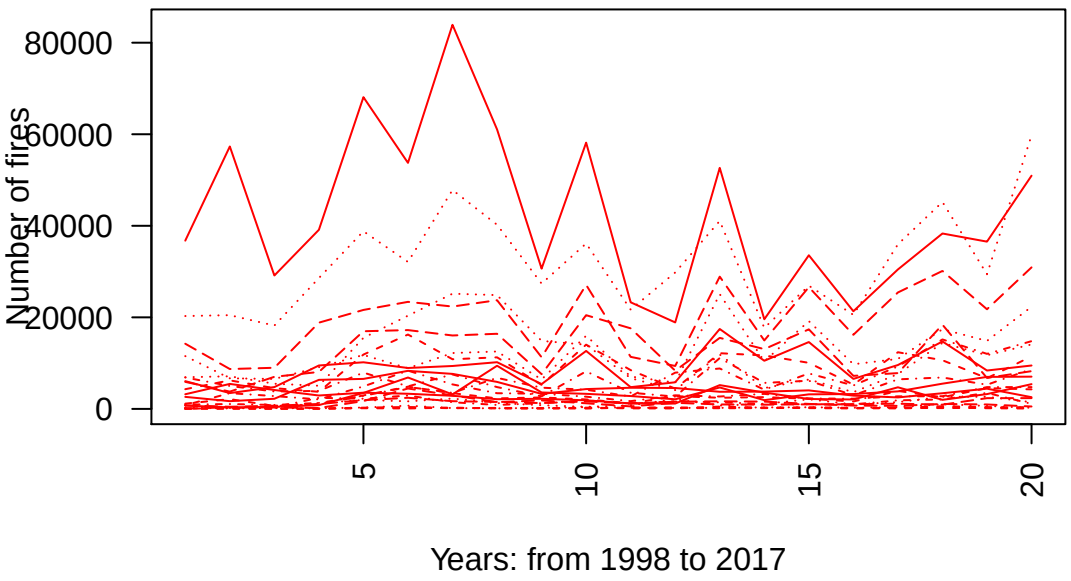
The dataset can be visualized in the below map. The intensity of the colour is proportional to the mean number of fires among the years 1998 - 2017, as displayed in the legend on the right.



In this dataset there is a big eterogeneity between the states. In the following boxplots, the mean and the variability between the number of fires in each state are displayed. The scale is the same for each state.



Moreover, in the following matplot every line corresponds to a different state. The y-axes represents the number of fires for that state while the x-axes represent the years (from 1998 to 2017). It can be seen that there is not a linear increasing of the number of fires among the years.



4 Methods

4.1 Choice of models

Because of the hierarchical structure of the dataset, building a hierarchical model was a natural choice. Since there are massive differences in variances and means between the states, a pooled model can not perform very well in analyzing a single state. Separate models however, can perform even better than hierarchical models since the number of states is quite high (23) and not all states are directly related because of the size of the country. Because of this, we have focused on comparing separate and hierarchical models.

First, a normal model is introduced, as it is a good basis to build further models on and it provides a good baseline for further results. Because of the aforementioned qualities of the dataset (mean and variance) the next model we built were a poisson and a negative binomial model, which should fit the dataset better. A separate and hierarchical versions of these models were implemented.

The negative binomial models proved to model our data better than their normal counterparts. With the poisson model we ran to an error with the relative effective sample size estimates for the likelihood (r_eff) that we could not solve even with the help of teaching assistants, and thus we could not properly compare this model to the other models.

A similar scenario occurred with a negative binomial regression model we built. The goal with this model was to use the hierarchical negative binomial model and extend it to handle the yearly data of the fires in each state and make predictions into the future using the available data. With this model as well, we ran into the same problem concerning likelihoods and thus were not able to perform coherent comparison with the working models.

The two models that we could not get working are appended to the end of the models section, but they are not compared with the other models because of the aforementioned reasons.

4.2 Choice of priors

We wanted to capture the large variance in the dataset with our choice of priors. For the normal hierarchical model we used a prior distribution $\mu_0 = N(\mu, \sigma)$, where μ is the mean of the dataset and σ is the variance of the dataset.

The priors for this model are not optimal, as they come from the data and give the models a "too good" idea on what to expect from the input data. The priors should not come directly from the data, instead they should be more general in order to help the model succeed better with unknown and upcoming data as well, not only the data it is tested with. But in this case with the normal model we decided to stick with these since no matter the priors, the normal models could not perform as well as the negative binomial models.

For the negative binomial model, as well as the regression model, we used the priors $\alpha = Exp(0.0006303)$ and $\beta = Exp(1.2)$.

The parameter of the exponential distribution of β was acquired by experimenting with different priors. The two parameters (of α and β) are related one to each other and the relationship is found by imposing the mean of a negative binomial equal to the mean of our dataset.

4.3 Methods for comparing the models

To compare the models with one another as well as the original data, we used posterior predictive checking. We used the fitted models to simulate new estimates for the number of fires in some states and then compared these draws to the actual data.

Additionally, we used the fitted models to predict the number of fires in each state for the last year of the dataset (2017) by leaving that year out during the fitting. Then we compared the prediction to the actual data. The plots for these comparisons can be found in section 8.

5 Definition of the models

The models built can be found from this section. First the normal models are introduced, after that the negative binomial models and lastly the Poisson and hierarchical regression model are presented, although they can not be compared to the other models.

5.1 Normal Models

The models (Separate and Hierarchical) presented in section 5.1 are all based on a normal prior.

5.1.1 Separate Normal Model

$$y_{ji} \sim N(\mu_j, \sigma_j)$$

```
# STAN CODE: SEPARATE NORMAL MODEL
separate_code = "

data {
  int<lower=0> N;              // number of data points
  int<lower=0> K;              // number of groups
  int<lower=1,upper=K> x[N];  // group indicator
  vector[N] y;
}

parameters {
  vector[K] mu;                // group means
  vector<lower=0>[K] sigma;    // group stds
```

```
}

model {
  y ~ normal(mu[x], sigma[x]);
}

generated quantities {
  vector[K] y_state;
  vector[N] log_lik;

  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[x[i]], sigma[x[i]]);

  for (i in 1:K)
    y_state[i]=normal_rng(mu[i], sigma[i]);
}
"
```

5.1.2 Hierarchical Normal Model

$$y_{ji} \sim N(\overline{\mu} + \mu_j, \sigma)$$

```
# STAN CODE: HIERARCHICAL NORMAL MODEL
hierarchical_code = "

data {
  int<lower=0> N;           // number of data points
  int<lower=0> K;           // number of groups
  int<lower=1,upper=K> x[N]; // group indicator
  vector[N] y;
}

parameters {
  real mu0;                 // prior mean
  real<lower=0> sigma0;     // prior std
  vector[K] mu;             // group means
  real<lower=0> sigma;      // common std
}

model {

  mu0 ~ normal(7933,25325);  // weakly informative prior
  sigma0 ~ cauchy(0,4);      // weakly informative prior
  mu ~ normal(mu0, sigma0);  // population prior with unknown parameters
  sigma ~ cauchy(0,4);       // weakly informative prior
  y ~ normal(mu[x], sigma);
}

generated quantities {
  real ypred;
  real mupred;
  vector[K] y_state;
  vector[N] log_lik;
```

```
  mupred = normal_rng(mu0,sigma0);
  ypred = normal_rng(mupred, sigma);

  for (i in 1:N)
    log_lik[i] = normal_lpdf(y[i] | mu[x[i]], sigma);

  for (i in 1:K)
    y_state[i]=normal_rng(mu[i], sigma);

}
"
```

5.2 Negative Binomial Models

As discussed in the methods section of this document, the data has a higher variance than mean. Because of this we applied a negative binomial model to the data.

5.2.1 Separate Negative Binomial Model

```
# STAN CODE: SEPARATE NEGATIVE BINOMIAL MODEL
separate_negative_bin = "

data {
  int<lower=0> N;             // number of data points
  int<lower=0> K;             // number of groups
  int<lower=1,upper=K> x[N];  // group indicator
  int<lower=0> y[N];
}

parameters {
  real<lower=0> alpha[K];
  real<lower=0> beta[K];
}

model {
  alpha ~ exponential(0.0006303);
  beta ~ exponential(1.2);
  y ~ neg_binomial(alpha[x], beta[x]);
}

generated quantities {
  int<lower=0> y_rep[K];
  vector[N] log_lik;

  for (i in 1:N)
    log_lik[i] = neg_binomial_lpmf(y[i] | alpha[x[i]], beta[x[i]]);

  for (i in 1:K)
    y_rep[i] = neg_binomial_rng(alpha[i], beta[i]);
}
"
```

5.2.2 Hierarchical Negative Binomial Model

```
 # STAN CODE: DEFINITION OF HIERARCHICAL NEGATIVE BINOMIAL MODEL
hierarchical_negative_bin = "

data {
  int<lower=0> N;           // number of data points
  int<lower=0> K;           // number of groups
  int<lower=1,upper=K> x[N]; // group indicator
  int<lower=0> y[N];
}

parameters {
  real alpha;
  real<lower=0> beta[K];
}

model {
  alpha ~ exponential(0.0006303);
  beta ~ exponential(1.2);
  y ~ neg_binomial(alpha, beta[x]);
}

generated quantities {
  int<lower=0> y_rep[K];
  vector[N] log_lik;

  for (i in 1:N)
    log_lik[i] = neg_binomial_lpmf(y[i] | alpha, beta[x[i]]);

  for (i in 1:K)
    y_rep[i] = neg_binomial_rng(alpha, beta[i]);
}
"
```

5.3 Hierarchical Poisson Model

Below is the definition for our Poisson model that we could not get working. The posterior and predictions it produces are reasonable, and the Rhat values are mostly converged, but the k values are out of bounds and r_{eff} contains NA values.

```
 # STAN CODE: DEFINITION OF HIERARCHICAL POISSON MODEL
poisson_h_model = "
data {
  int<lower=0> N;
  int<lower=0> K;           // number of groups
  int  xx[N];        // predictor (year)
  vector[N] x;        // predictor (year)
  int<lower=0> y[N];  // response (n of fires)
}
parameters {
  vector[K] alpha;
}
```

```
model {
  for (i in 1:K) {
    alpha[i] ~ normal(0,100); // Bad prior
  }

  for( i in 1:N) {
    y[i] ~ poisson(alpha[xx[i]]);
  }
}

generated quantities {
  int<lower=0> ypred[K];
  vector[N] log_lik;
  for (i in 1:K)
    ypred[i] = poisson_rng(alpha[i]);

  for (i in 1:N)
    log_lik[i] = poisson_lpmf(y[i] | alpha[xx[i]]);
}
"
```

5.4 Hierarchical Regression Model

The regression model shares many charasteristics with the Poisson model above. It also provides predictions that make sense for the state in question but once again r_{eff} contains NA values and thus the model can not be compared in a meaningful way.

```
 # STAN CODE: DEFINITION OF HIERARCHICAL REGRESSION MODEL
regression_model = "

data {
  int<lower=0> N;     // number of datapoints
  int<lower=0> K;     // number of groups
  int  xx[N];        // predictor (year)
  vector[N] x;        // predictor (year)
  int<lower=0> y[N];  // response (n of fires)
  real xpred;         // predictor
}
parameters {
  vector[K] alpha;
  vector[K] beta;
  real phi;
}

model {
  phi ~ exponential(0.1);
  alpha ~ normal(0,100);
  beta ~ normal(0,100);

  for( i in 1:N) {
    y[i] ~ neg_binomial_2(alpha[xx[i]] + beta[xx[i]] * x[i], phi);
  }
}
```

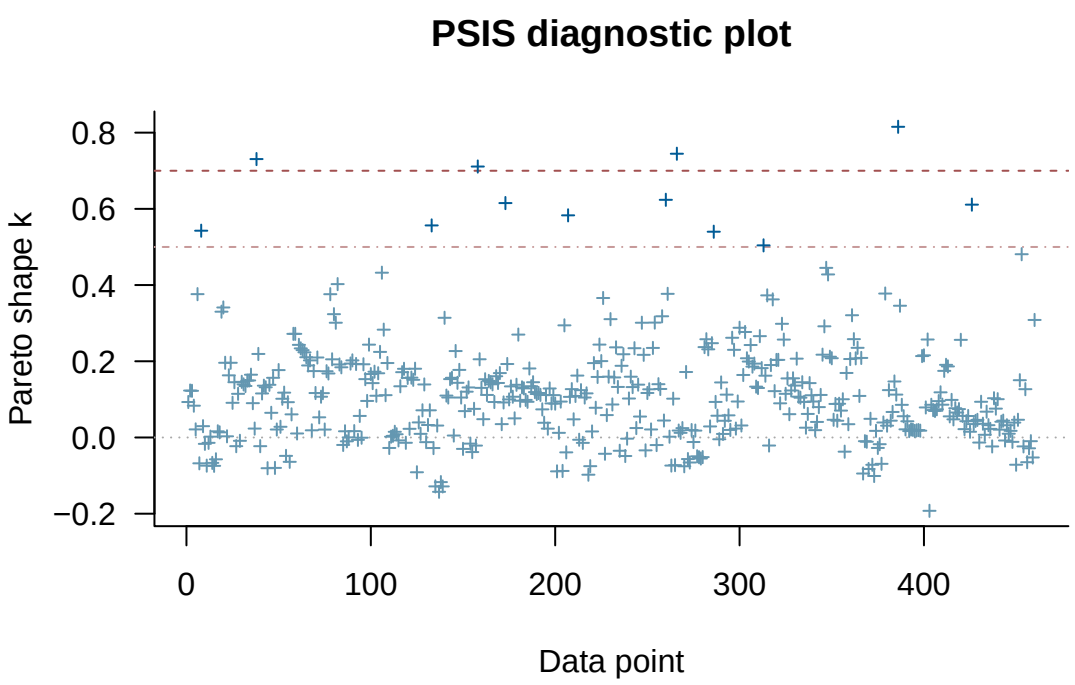
```
generated quantities {
  int<lower=0> ypred[K];
  vector[N] log_lik;
  for (i in 1:K)
    ypred[i] = neg_binomial_2_rng(alpha[i] + beta[i] * xpred, phi);

  for (i in 1:N)
    log_lik[i] = poisson_log_lpmf(y[i] | alpha[xx[i]] + beta[xx[i]]);
}
"
```

6 Results

Here the PSIS-LOO elpd values and the k-values for each of the two normal models and the two negative binomial models introduced in the last section as well as the effective number of parameters Peff for each of the model. The values for PSIS_LOO and the peff are displayed under the plot of k-values.

SEPARATE NORMAL MODEL

```
##
## Computed from 4000 by 460 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -4088.2 34.9
## p_loo        45.1  5.3
## looic      8176.4 69.8
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     448   97.4%   692
##  (0.5, 0.7]   (ok)         8    1.7%   272
##    (0.7, 1]   (bad)        4    0.9%   44
##    (1, Inf)   (very bad)   0    0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

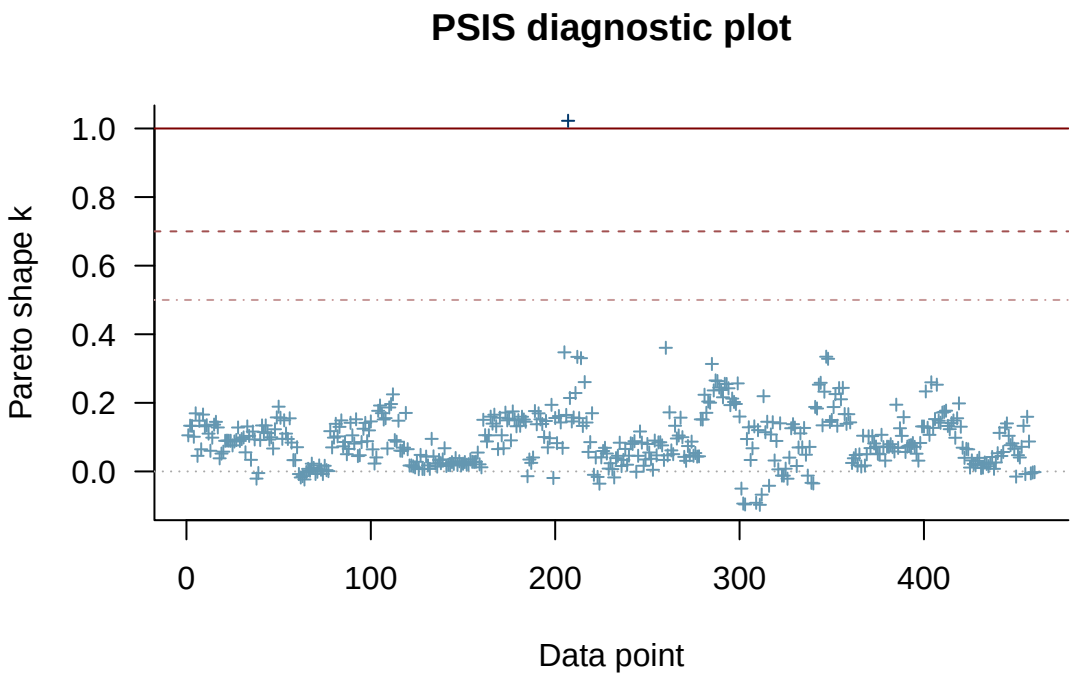



```
## [1] -4088.191
```

```
## [1] 45.06843
```

HIERARCHICAL NORMAL MODEL

```
##
## Computed from 4000 by 460 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -4621.9 45.0
## p_loo        29.5  7.7
## looic      9243.7 90.0
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     459   99.8%   365
##  (0.5, 0.7]   (ok)         0    0.0%   <NA>
##    (0.7, 1]   (bad)        0    0.0%   <NA>
##    (1, Inf)   (very bad)   1    0.2%   30
## See help('pareto-k-diagnostic') for details.
```

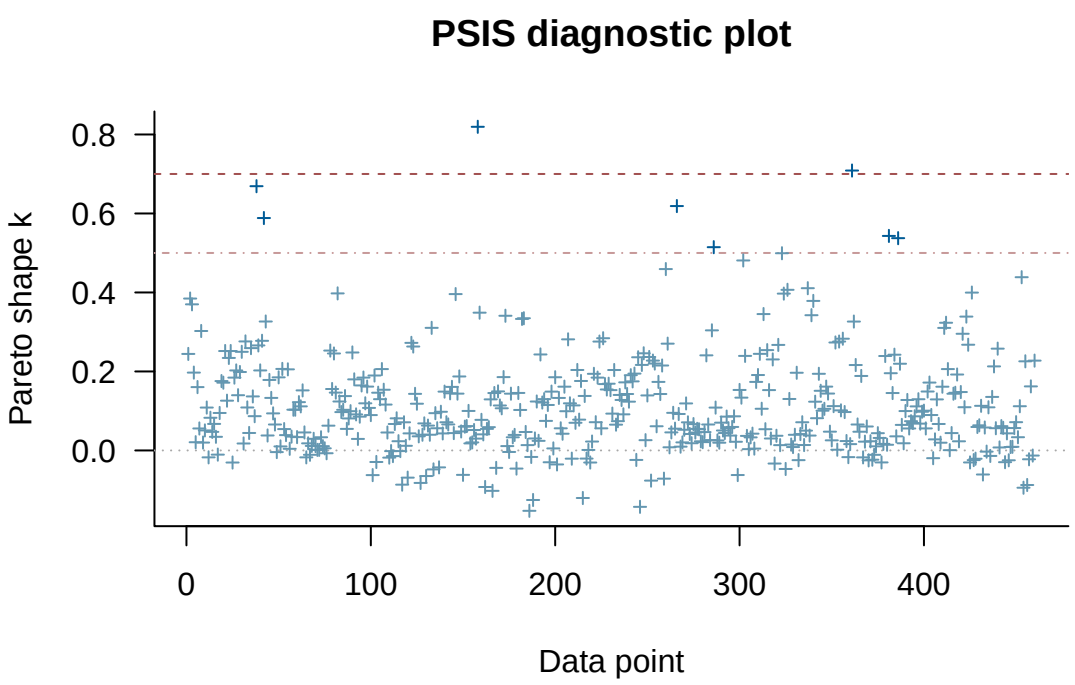



```
## [1] -4621.873
```

```
## [1] 29.53274
```

SEPARATE NEGATIVE BINOMIAL MODEL

```
##
## Computed from 4000 by 460 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -4053.2 36.0
## p_loo        49.8  4.7
## looic      8106.5 71.9
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     452   98.3%   747
##  (0.5, 0.7]   (ok)         6    1.3%   157
##    (0.7, 1]   (bad)        2    0.4%   51
##    (1, Inf)   (very bad)   0    0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

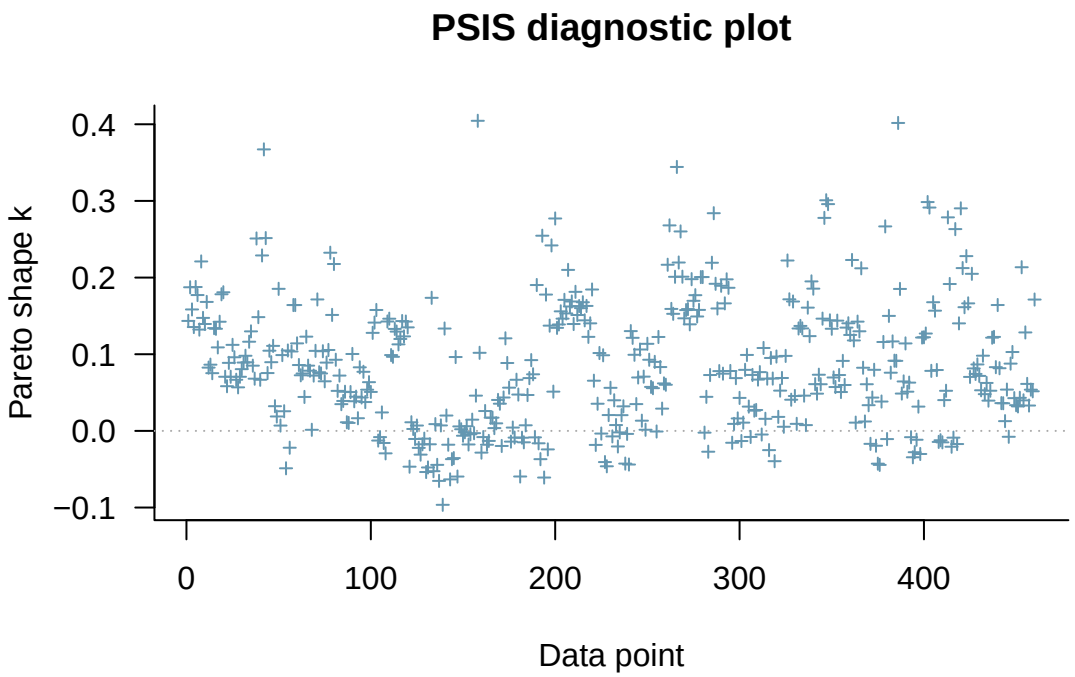



```
## [1] -4053.233
```

```
## [1] 49.82759
```

HIERARCHICAL NEGATIVE BINOMIAL MODEL

```
##
## Computed from 4000 by 460 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -4063.0 36.3
## p_loo        24.5  2.1
## looic      8126.0 72.7
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```




```
## [1] -4053.233
```

```
## [1] 24.48166
```

7 Choosing the best model

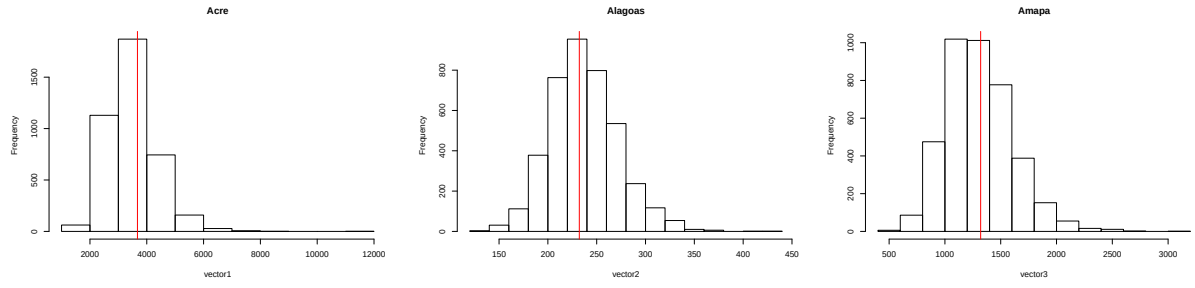
The best model is chosen by how closely the fitted model resembles the actual data and the accuracy of its predictions for individual states, as well as the PSIS_LOO values visible in the previous section. Separate Negative Binomial model is the best one between the first four models. For this model, the fit is done on the years 1998-2016 and then the prediction for the year 2017 is compared to the data.

8 Analysis of the results

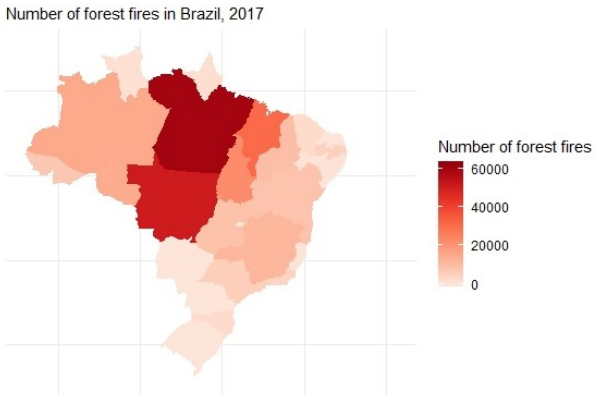
In this section we perform comparison between our most accurate model and the original data using the methods described in section 4.3.

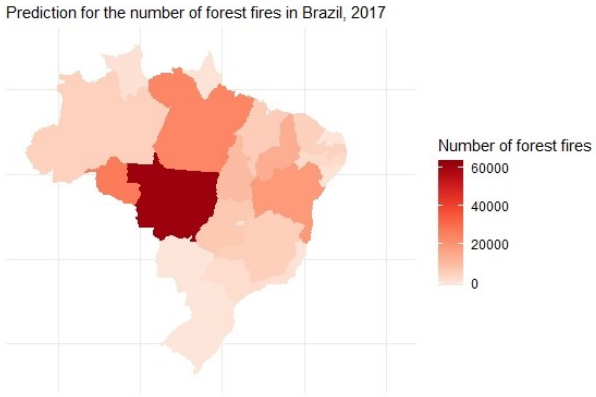
8.1 Separate Negative binomial model

POSTERIOR PREDICTIVE CHECKING FOR THE FIRST THREE STATES



COMPARISON BETWEEN PREDICTION AND REAL DATA (2017)





9 Discussion of the results, of the problems and potential improvements

As can be seen from the plots in the previous section, our best model can estimate the number of fires some what accurately, but there are small differences in most of the states. The overalls trends are correct and the distributions for individual states are reasonable and resemble the originals.

The largest problem we faced during this project, as well as the most potential improvement is the Poisson model we could not get working. Because negative binomial has to do with "the number of failures" in an experiment, which is an element that is not present in our dataset since it is impossible to track the fires that did not develop, Poisson model has great potential in being able to model the data better. The regression model should also be considered as a potential improvement, and with fine tuned priors it could provide better estimates than our current best model.

Moreover, the original dataset presents the distribution of the forest fires among the months and the exact day of the fire. This information about the time series are not taken into account in the above proposed model, but should be included for a more sophisticated analysis.

To conclude, the number of forest fires is not trivial to estimate only by looking at the number of past fires. The phenomena has multiple different factors that should also be incorporated into the predictive models if one wants to perform meaningful analysis. The temperatures, winds and amount of rainfall as well as the local population and legistlation are some of the important factors that have a huge effect on how the number of fires develop over the years. Our model that only analyzes the amount of past fires can provide helpful

estimates that are in the right ballpark for the individual states, but more sophisticated analysis is needed to truly be able to provide an accurate estimate.

10 References

Dataset:

- https://www.kaggle.com/gustavomodelli/forest-fires-in-brazil

Other links used:

- https://datascienceplus.com/bayesian-regression-with-stan-beyond-normality/
- https://mc-stan.org/docs/2_20/functions-reference/nbalt.html
- https://mc-stan.org/loo/reference/loo-glossary.html
- https://github.com/avehtari/BDA_R_demos/blob/master/demos_rstan/ppc/poisson-simple.stan