Final_Yi

2018-12-03

Model Extension: CAR & Zero Inflation

A linear binormial regression analogy

Lik what we did in the baseline model. We process by treating the underlying deterministic model as providing an expected default times for each city around which there will be variation due to both measurement error and simplifications. Consider the typical formulation of a linear regression, where y_n is the is an observable default time, x_n is a row vector of unmodeled predictors (independent variables), β is a coefficient vector parameter and we separate the intercep as a. In the city level, we assume that the number of individual records in city i is n_i.Thus, we have the model:

$$y_i \sim Binormial(n_i, logit^{-1}(a + x_i\beta))$$

As a robust prior distribution option, we take $\beta \sim cauchy(location = 0, scale = 2.5)$.

In model 2, we will extend this model in two ways: (1) incoperate the geographic state information (2) zero inflated model.

Extension one: Incoperate Geographic Information

In this project, we utilize the IAR prior for state feature. Intrisnic conditional autoregressive (IAR) is an extension of conditional autoregressive (CAR) models, which are popular as prior distributions for spatial random effects with areal spatial data. In our model, we have a random quantity $\phi = (\phi_1, \phi_2, \ldots, \phi_{32})$ at 32 state areal locations. In each state, we have the individual records aggregated at the city level. And each city data belong to one state. According to the Brook's Lemma, the joint the distribution of ϕ can be expressed as the followings:

$$\phi \sim N(0, [D_\tau(I - \alpha B)]^{-1})$$

In this formula, we have:

- $D = diag(m_i)$ is an 32×32 diagonal matrix with m_i is the number of the neighbors for the state i
- $D_\tau = \tau D$ and τ is the hyperparameter in the conditional distributions of the ϕ
- α is the parameter that controls spatial dependence. In IAR, we let $\alpha = 1$
- $B = D^{-1}W$ is the scaled adjacency matrix. And W is the adjacency matrix. ($w_{ii} = 0$,$w_{ij} = 1$ if the state i is a neighbor of state j , and $w_{ij} = 1$ otherwise)

We can simplifies the IAR model to:

$$\phi \sim N(0, \tau(D - W)]^{-1})$$

In IAR model, we have a singular precision matrix and an improper prior distribution. However, in practice, IAR models are fit with a sum to zero constrains: $_{i} _i = 0 $ for each connected component of the graph. In this way, we can interpret both overall means and the component-wise means.

Through log probability accumulator, we can accure computational efficiency gains. We have:

$$log(p(\phi|\tau)) = -\frac{n}{2}log(2\pi) + \frac{1}{2}log(det^*(\tau(D-W))) - \frac{1}{2}\phi^T\tau(D-W)\phi$$

$$= -\frac{n}{2}log(2\pi) + \frac{1}{2}log(\tau^{n-k}) + \frac{1}{2}log(det^*(\tau(D-W)) - \frac{1}{2}\phi^T\tau(D-W)\phi$$

In this formula, $det^*(A)$ is the generlized determinant of the square matrix A defined as the product of its non-zero eigenvalues, and the k is the number of the connented component in the graph.(k=1 for our data) Dropping the additive constants, the qunantity to increment becomes:

$$\frac{1}{2}log(\tau^{n-k}) - \frac{1}{2}\phi^T\tau(D-W)\phi$$

In our model, we assume the hyperparameter $\tau \sim Gamma(shape = 2, rate = 2)$. We define the sparse_iar_lpdf function as a more efficient spare representation as the following:

```
functions {
  real sparse_iar_lpdf(vector phi, real tau, int[,] W_sparse, vector D_sparse, vector la
mbda, int S, int W_n) {
      row_vector[S] phit_D; // phi' * D
      row_vector[S] phit_W; // phi' * W
      vector[S] ldet_terms;

      phit_D = (phi .* D_sparse)';
      phit_W = rep_row_vector(0, S);
      for (i in 1:W_n) {
        phit_W[W_sparse[i, 1]] = phit_W[W_sparse[i, 1]] + phi[W_sparse[i, 2]];
        phit_W[W_sparse[i, 2]] = phit_W[W_sparse[i, 2]] + phi[W_sparse[i, 1]];
      }

      return 0.5 * ((S-1) * log(tau)
                    - tau * (phit_D * phi - (phit_W * phi)));
  }
}
```

After we get the IAR prior, we can take it into our model.For the i the city in the j the state, we have:

$$y_{ij} \sim Binormial(n_{ij}, logit^{-1}(a + \phi_j + x_{ij}\beta))$$

* Reason for design the model in this way rather than hierarchical model:

1. Hierarchical extension will greatly expend the demension of the parameter space. Consequently, the estimation convergence would be much more difficult.Just focus on the binormial regression part. If we take the hierarchical extension both on the intercept and coefficient terms, there will have $32 \times 1 = 32$ intercept paramters and $32 \times 7 = 224$ coefficient parameters (total 256 parameters).For taking the hierarchical extension on βs, we have 1 intercept and $7 \times 32 = 224$ coefficient in the binormial regression model (total 225 parameters). But now, we only have 1 parameter for overall intercept, 32 parameter for the state level effecr and 7 parameter for the different independent variables' effects.

2. This design of model is better for interpreation and better for us to solve our research problems.Now the coefficient terms are not depend on the state prior. Thus, we can estimate the overall effect from these independent variables, like age and income. And on the state level effect, we can have the overall idea based on the estimated parameter values.

Extension two: zero inflated model

Zero-inflated model originally provide mixture of a mixtures of a Poisson and Bernoulli probability mass function to allow more flexibility in modeling the probability of a zero outcome. Zero-inflated models, as defined by Lambert (1992), add additional probability mass to the outcome of zero. But, this extension can also be applied for other categorical distributions like binomial distribution we used in this project.

We assume a parameter θ as the probability of drawing a zero and the probability $1-\theta$ as drawing fro mthe Binormal distribution. The prior distribution of θ is uniform between 0 and 1, since we have no extra information about this parameter. The distribution function is thus:

$$p(y_n|\theta, a, \beta) = \begin{cases} \theta + (1-\theta) \times Binomial(0|a, \beta, \phi) & y_n = 0 \\ (1-\theta) \times Binomial(0|a, \beta, \phi) & y_n > 0 \end{cases}$$

In stan, we estimate the model in this following ways:

```
for (j in 1:N_train){
    if (y[j] == 0){
      target += log_sum_exp(bernoulli_lpmf(1 | theta), bernoulli_lpmf(0 | theta) + binom
ial_logit_lpmf(y[j] | n_city_train[j],alpha[state_train[j]] + X_train[j,]* beta));
      }
    else{
      target += bernoulli_lpmf(0 | theta) + binomial_logit_lpmf(y[j] | n_city_train[j],a
lpha + X_train[j,] * beta);}
  }
```

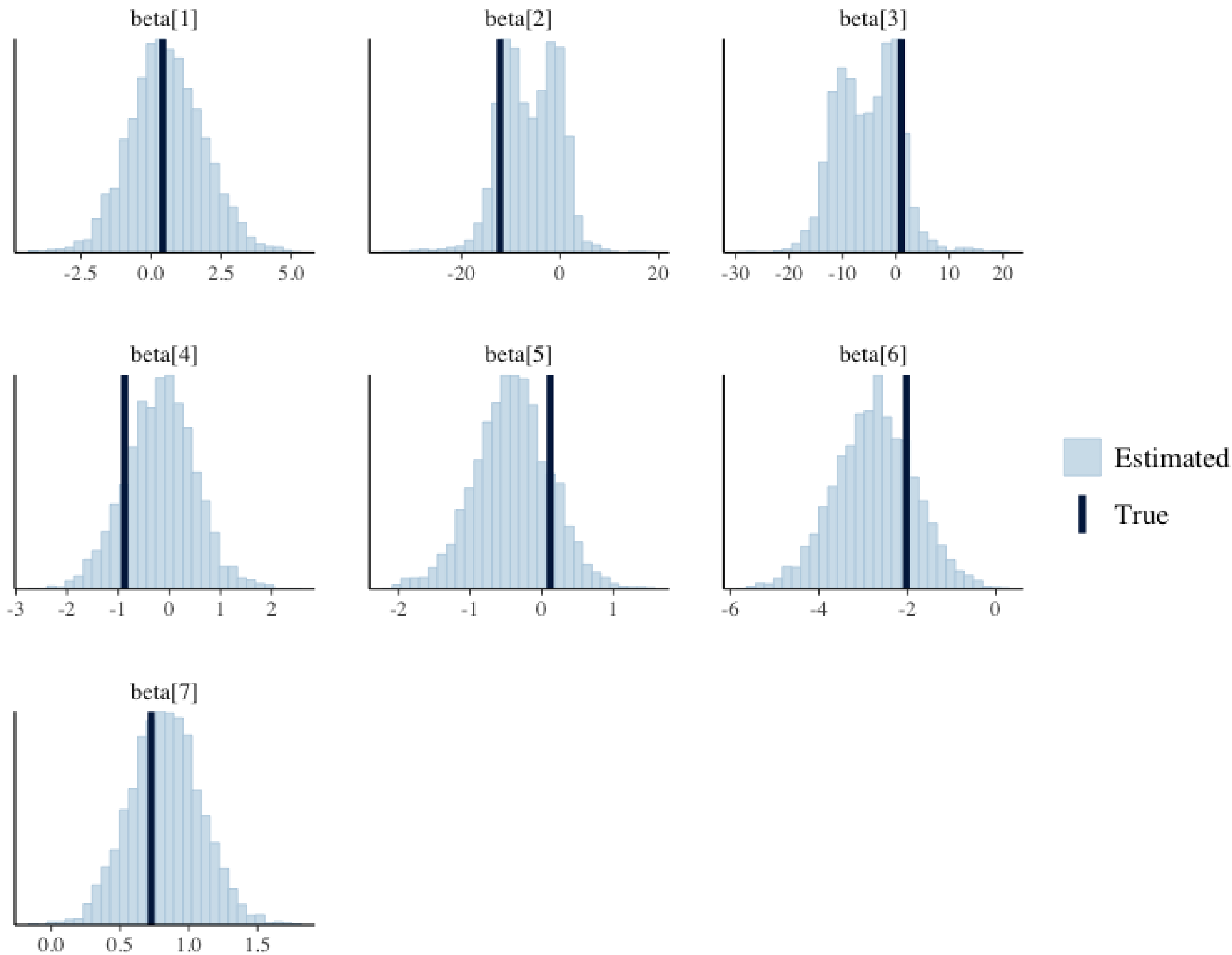
And we predict the y_{rep} in the following way:

```
generated quantities{
  int y_rep[N_train];
  real<lower =0,upper=1> zero_train[N_train];
  for (i in 1:N_train){
    zero_train[i] = uniform_rng(0,1);
    if (zero_train[i] < theta){
      y_rep[i] = 0;
      }
    else{
      y_rep[i] = binomial_rng(n_city_train[i],inv_logit(alpha[state_train[i]] + X_train
[i,]* beta));
      }
  }
}
```

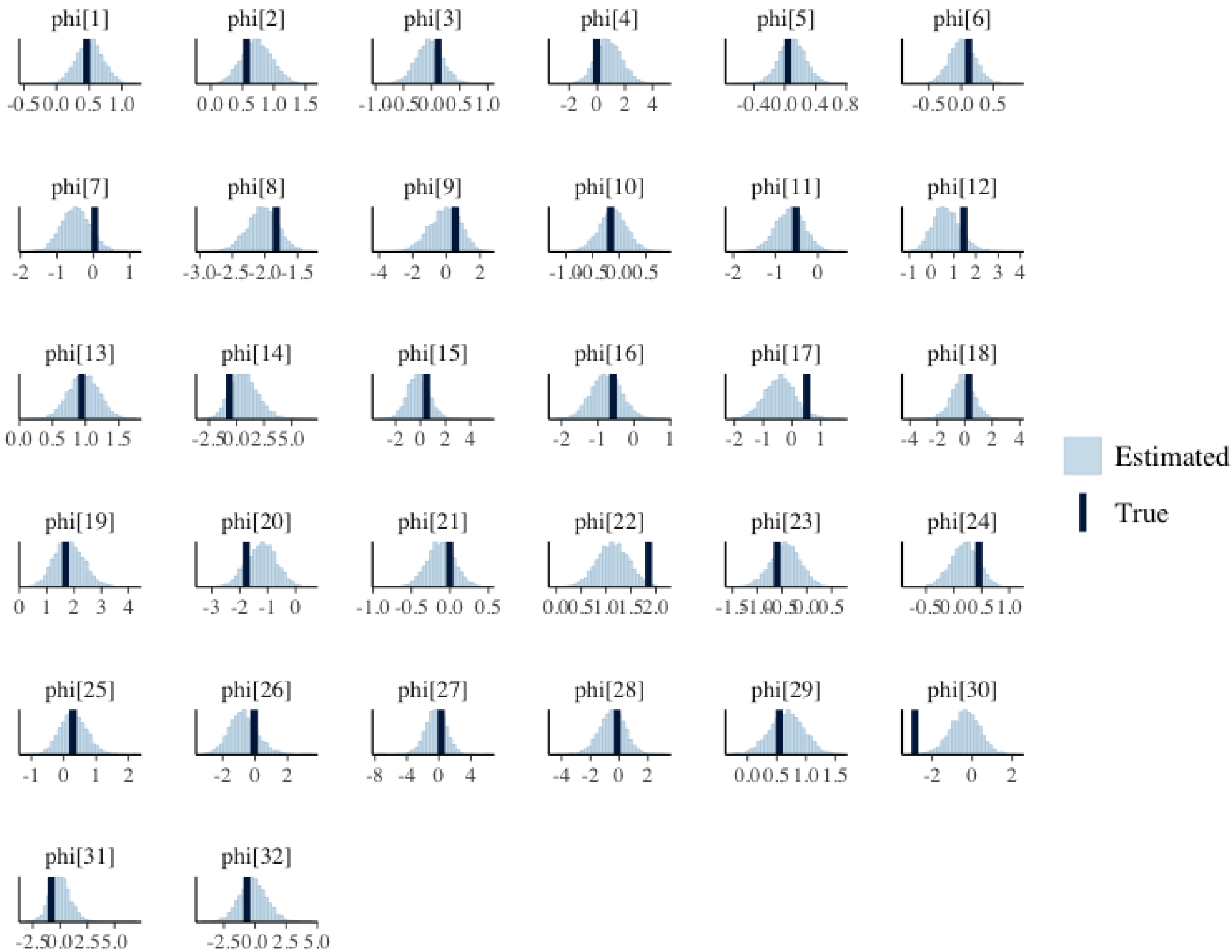
Model Coveage testing wit fake data.

In this part, we first simulate the fake data as we assume in thie model. Then, we will check that our model works well with the data that we have simulated ourselves. In this following are the model coverage plots.

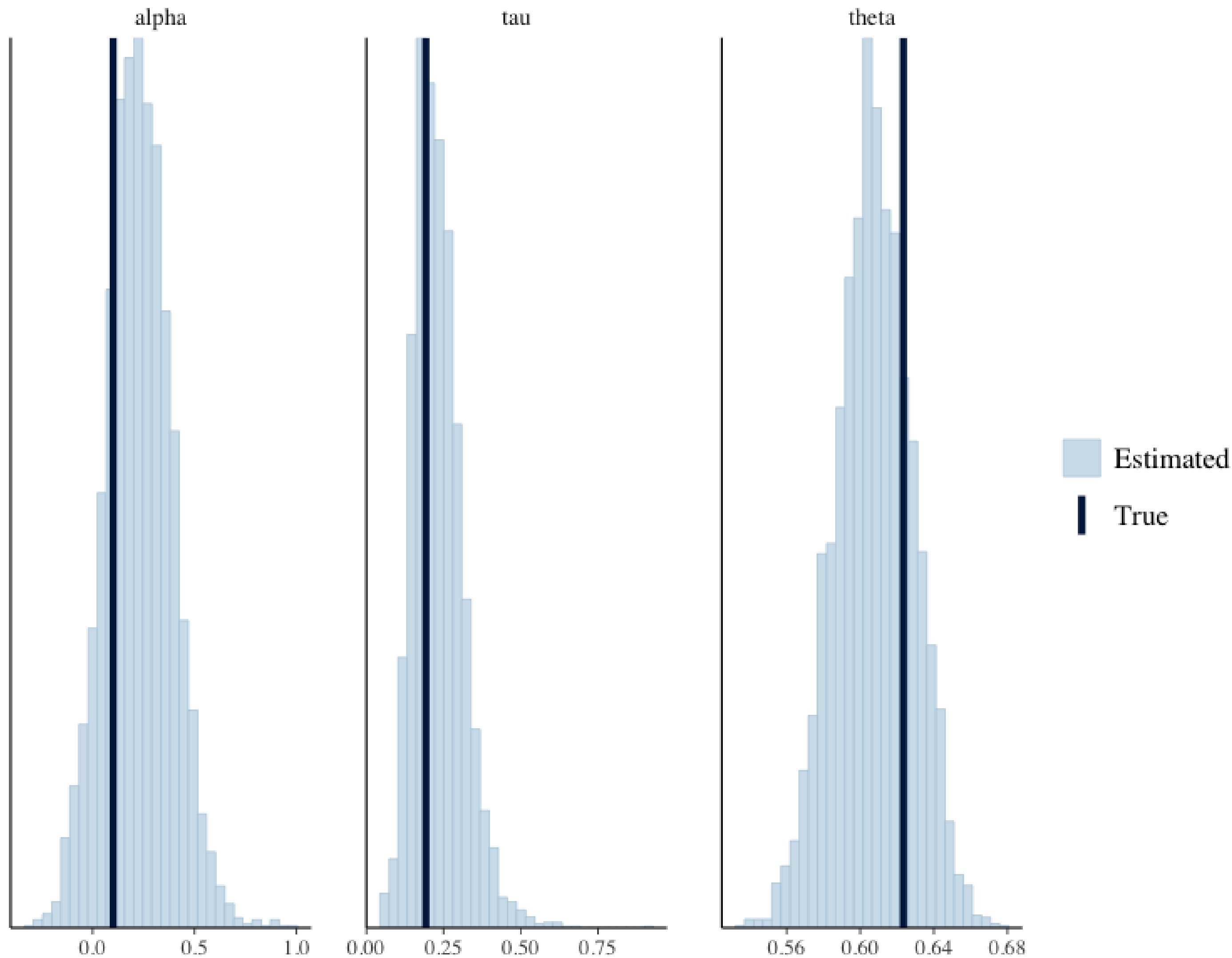
We now assessing the parameter recoverage of β parameters.



In the plot plot is the parameter recoverage of ϕ parameters.



Finally, let's check the parameter recoverage of α, θ, τ

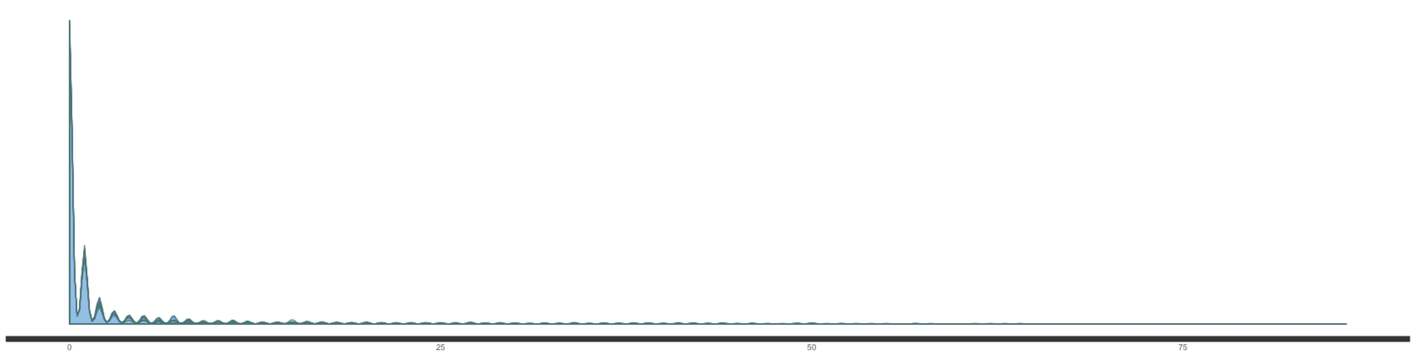


As we can see, all the parameters in our fake data recover very well. This means it is reliable to use rstan to run this model.

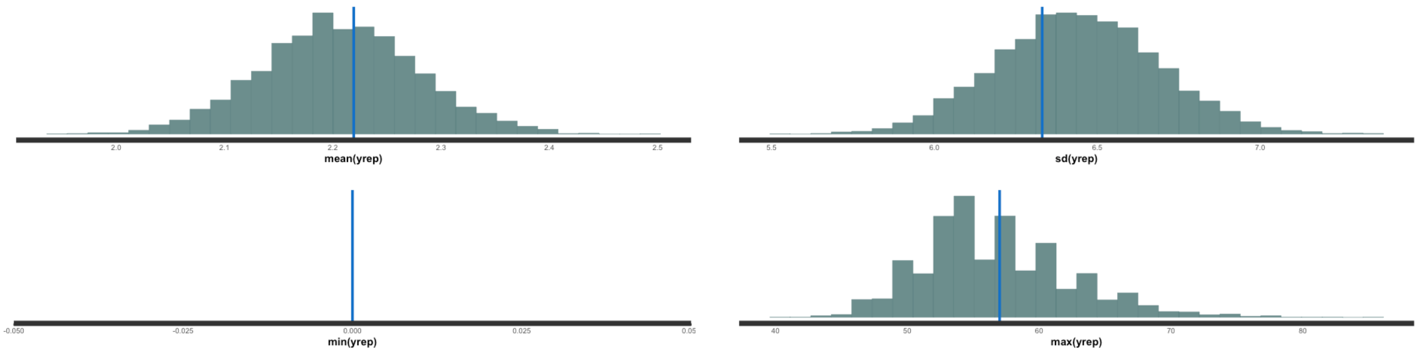
Model Check

Posterior Predict Check (PPC)

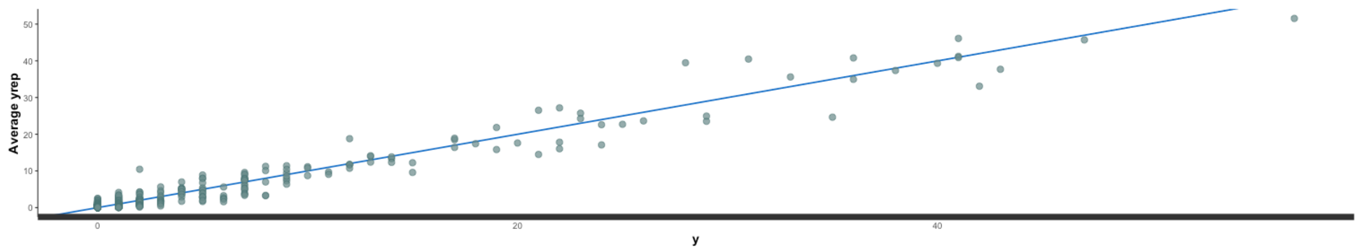
In the plot below we have the kernel density estimate of the observed data (y, thicker curve) and 200 simulated data sets (y_{rep}, thin curves) from the posterior predictive distribution. If the model fits the data well, as it does here, there is little difference between the observed dataset and the simulated datasets.



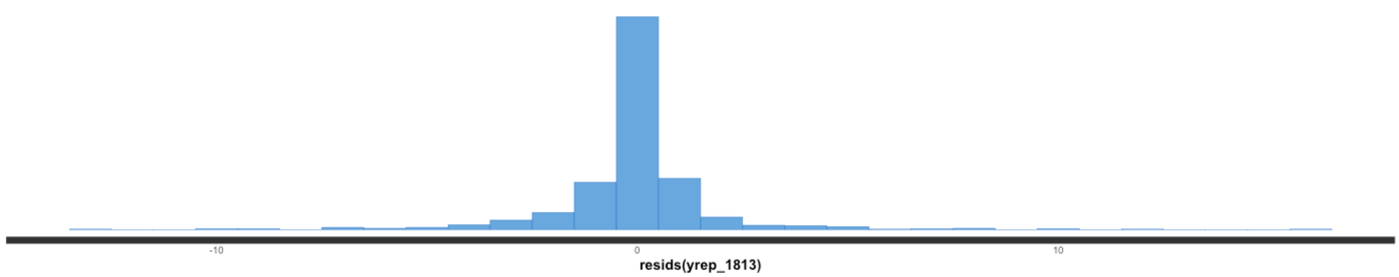
As we can see from the polt below, y_{rep} behavior well in the four most common statistics. Ideally this vertical line would fall somewhere within the histogram, as what we did.



The plot below shows the observed and average simulated value. As we can see the model fit the data very well without obvious outliers.



The residuals centered at 0 and have small variance. This indecates that the model fit is acceptable.



Cross Validation & MSE

In order to determine our model performance, again we do the 5-floder cross validation. And we calculate the MSE for each training dataset with our model. And then we get the average MSE. The stan code we used to simulate the $y_h at$ is as the following:

```
generated quantities{
  int y_rep_cv[N_test];
  real<lower =0,upper=1> zero_test[N_test];
  for (i in 1:N_test){
    zero_test[i] = uniform_rng(0,1);
    if (zero_test[i] < theta){
      y_rep_cv[i] = 0;
      }
    else{
      y_rep_cv[i] = binomial_rng(n_city_test[i],inv_logit( alpha[state_test[i]] + X_test
[i,]* beta));
      }
  }
}
```

Acrroding to the result, the baseline MSE is 128164. But for our model, the average MSE is **7583** with the standard deviation **6831**. Thus, we can say that our model have a huge improve from the baseline.

Model result

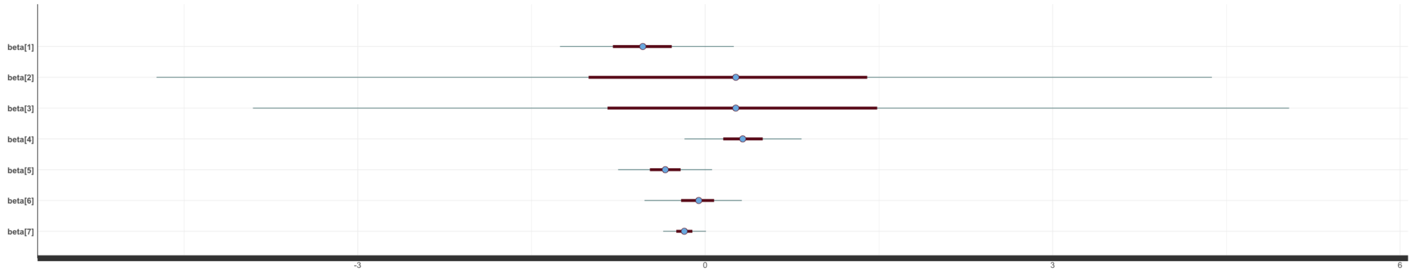
Brief results

In the following is the basic model results:

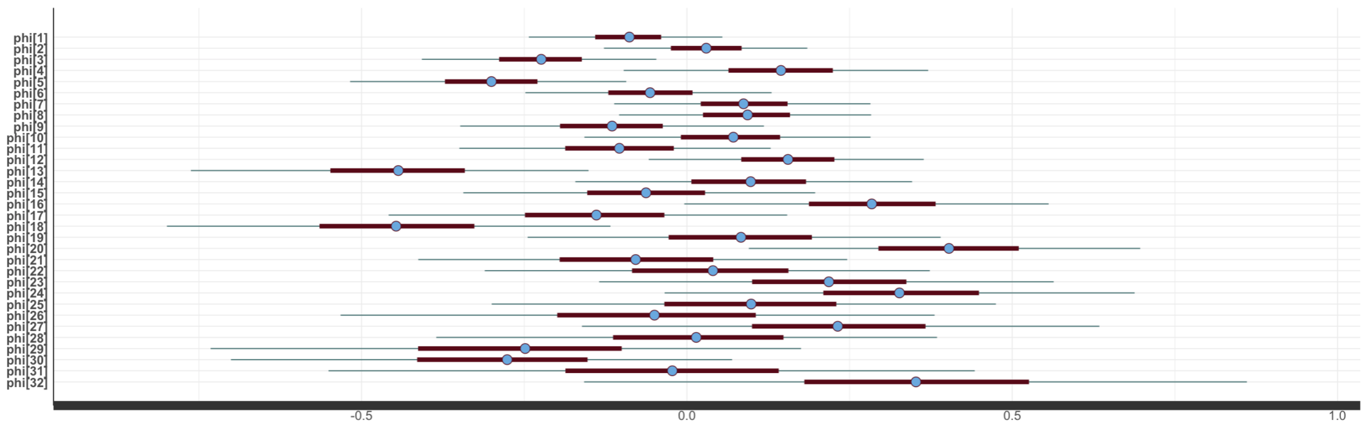
	mean	se_mean	sd	2.5%	25%	50%	75%	98%	n_eff	Rhat
alpha	-2.67	0.00	0.04	-2.76	-2.70	-2.67	-2.64	-2.59	187	1.0
beta[1]	-0.53	0.03	0.38	-1.25	-0.80	-0.54	-0.29	0.25	143	1.0
beta[2]	0.19	0.23	2.11	-4.74	-1.01	0.26	1.40	4.38	86	1.1
beta[3]	0.33	0.22	2.09	-3.91	-0.84	0.26	1.48	5.04	94	1.0
beta[4]	0.32	0.03	0.25	-0.18	0.16	0.32	0.50	0.83	97	1.1
beta[5]	-0.34	0.02	0.20	-0.75	-0.48	-0.34	-0.21	0.06	109	1.1
beta[6]	-0.07	0.02	0.21	-0.52	-0.21	-0.06	0.08	0.32	134	1.0
beta[7]	-0.18	0.01	0.10	-0.36	-0.25	-0.18	-0.11	0.01	217	1.0
phi[1]	-0.09	0.00	0.08	-0.24	-0.14	-0.09	-0.04	0.05	852	1.0
phi[2]	0.03	0.00	0.08	-0.13	-0.02	0.03	0.08	0.19	2025	1.0
phi[3]	-0.23	0.00	0.09	-0.41	-0.29	-0.22	-0.16	-0.05	1803	1.0
phi[4]	0.14	0.00	0.12	-0.10	0.06	0.14	0.22	0.37	1829	1.0
phi[5]	-0.30	0.00	0.11	-0.52	-0.37	-0.30	-0.23	-0.09	3299	1.0
phi[6]	-0.06	0.00	0.10	-0.25	-0.12	-0.06	0.01	0.13	2359	1.0
phi[7]	0.09	0.00	0.10	-0.11	0.02	0.09	0.15	0.28	2368	1.0
phi[8]	0.09	0.00	0.10	-0.10	0.02	0.09	0.16	0.28	2423	1.0
phi[9]	-0.12	0.00	0.12	-0.35	-0.20	-0.12	-0.04	0.12	3345	1.0
phi[10]	0.07	0.00	0.11	-0.16	-0.01	0.07	0.14	0.28	2464	1.0
phi[11]	-0.10	0.00	0.12	-0.35	-0.19	-0.10	-0.02	0.13	4003	1.0
phi[12]	0.16	0.00	0.11	-0.06	0.08	0.16	0.23	0.36	2948	1.0
phi[13]	-0.45	0.00	0.15	-0.76	-0.55	-0.44	-0.34	-0.15	3698	1.0
phi[14]	0.09	0.00	0.13	-0.17	0.01	0.10	0.18	0.35	3833	1.0
phi[15]	-0.06	0.00	0.14	-0.34	-0.15	-0.06	0.03	0.20	3366	1.0
phi[16]	0.28	0.00	0.14	0.00	0.19	0.28	0.38	0.56	2951	1.0
phi[17]	-0.14	0.00	0.16	-0.46	-0.25	-0.14	-0.03	0.15	4033	1.0
phi[18]	-0.45	0.00	0.18	-0.80	-0.56	-0.45	-0.33	-0.12	1581	1.0
phi[19]	0.08	0.00	0.16	-0.24	-0.03	0.08	0.19	0.39	4147	1.0
phi[20]	0.40	0.00	0.16	0.10	0.29	0.40	0.51	0.70	1102	1.0
phi[21]	-0.08	0.00	0.17	-0.41	-0.20	-0.08	0.04	0.25	4140	1.0
phi[22]	0.04	0.00	0.18	-0.31	-0.08	0.04	0.16	0.37	4044	1.0
phi[23]	0.22	0.00	0.18	-0.13	0.10	0.22	0.34	0.56	2248	1.0
phi[24]	0.33	0.00	0.18	-0.03	0.21	0.33	0.45	0.69	3626	1.0
phi[25]	0.10	0.00	0.20	-0.30	-0.03	0.10	0.23	0.47	3735	1.0
phi[26]	-0.05	0.00	0.23	-0.53	-0.20	-0.05	0.11	0.38	3186	1.0
phi[27]	0.23	0.00	0.20	-0.16	0.10	0.23	0.37	0.63	2633	1.0
phi[28]	0.01	0.00	0.20	-0.39	-0.11	0.01	0.15	0.38	3371	1.0
phi[29]	-0.26	0.00	0.23	-0.73	-0.41	-0.25	-0.10	0.18	2812	1.0
phi[30]	-0.29	0.00	0.20	-0.70	-0.41	-0.28	-0.15	0.07	1902	1.0
phi[31]	-0.03	0.00	0.25	-0.55	-0.19	-0.02	0.14	0.44	3394	1.0
phi[32]	0.35	0.00	0.26	-0.16	0.18	0.35	0.53	0.86	3258	1.0
theta	0.01	0.00	0.01	0.00	0.00	0.01	0.01	0.03	213	1.0
tau	2.49	0.05	0.84	1.23	1.88	2.37	2.98	4.46	240	1.0
lp__	-812.85	0.20	4.82	-823.32	-815.77	-812.58	-809.53	-804.33	593	1.0

A quick check for the Rhat in our model is all very good. The posterior confidence interval of the parameters are show as the following plots.

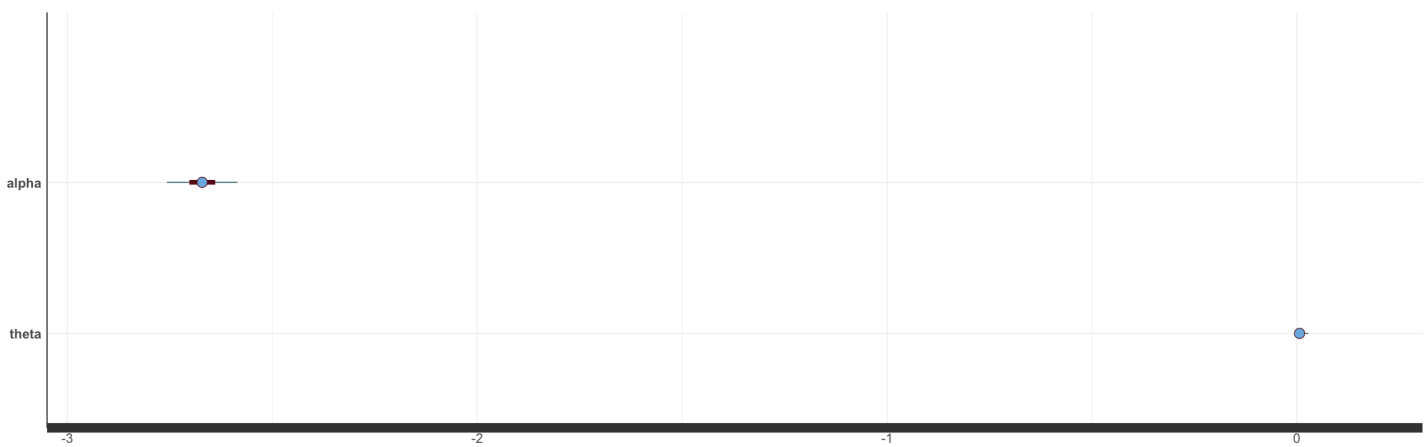
Confidence intervial and interpretation



As we can see above, the effect from the age is significant based on the 95% confidence interval. Other parameter are not significant enough.



The state effects are obvious. We can see that ϕ_3 (QUERETARO DE ARTEAGA), ϕ_5 (GUANAJUATO) , ϕ_{13} (AGUASCALIENTES) and ϕ_{18} (VERACRUZ LLAVE) have the nigative effect, which means these state is less likely to have default. However, ϕ_{20} (MICHOACAN DE OCAMPO) has a significantly postive effect.



As we can see, the overall offset effect is obvious that for about -2.67. And on avrage, there will have 1% of cities have no default at all. On 95% confidence interval, there will have less than 3% of cities have no default.