

# Generalized linear model: Bioassay with Stan

November 4, 2018

The Stan model is defined as:

---

```
data {  
  int animals[4];  
  int deaths[4];  
  vector[4] doses;  
  vector[2] mean_val;  
  matrix[2, 2] covariance;  
}  
  
parameters {  
  vector[2] alpha_beta;  
}  
  
model {  
  alpha_beta ~ multi_normal(mean_val, covariance);  
  deaths ~ binomial_logit(animals, alpha_beta[1] + alpha_beta[2] * doses);  
}
```

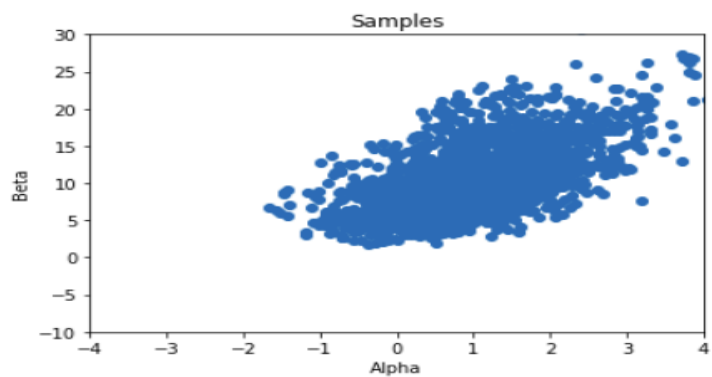
---

The  $\hat{R}$  values for  $\alpha$  and  $\beta$  are 1.002 and 1.003 accordingly.  
Since  $\hat{R}$  values are close to 1, we can say that the model **has converged**.

Stan also provides a summary of the results.  
Below is the summary of the experiment.

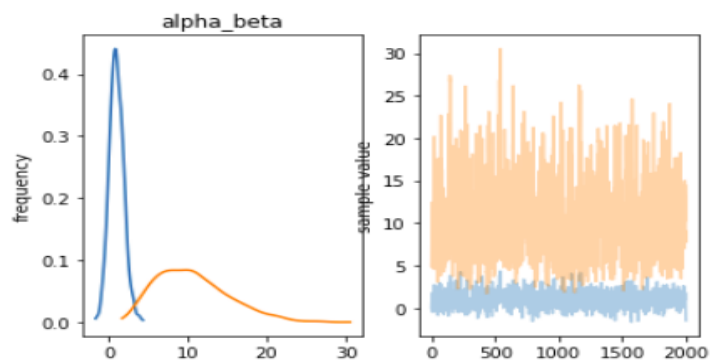
	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
alpha_beta[1]	1.001876	0.032972	0.909921	-0.707765	0.377217	0.957580	1.596745	2.939366	761.575357	1.002352
alpha_beta[2]	10.586120	0.201969	4.589509	3.491286	7.050669	10.100277	13.364230	21.016178	516.370865	1.003996
lp__	-7.132224	0.040386	1.020986	-9.831832	-7.531853	-6.824316	-6.394639	-6.111095	639.115474	0.999760

Below is the scatterplot of the draws.



Stan also provides useful visualizations.

Below is a visualization of  $\alpha$  and  $\beta$  provided by Stan.



---

```

import pystan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

corr = 0.5
sigma_alpha = 2
sigma_beta = 10
mu_alpha = 0
mu_beta = 10

animals = np.array([5, 5, 5, 5])
doses = np.array([-0.86, -0.30, -0.05, 0.73])
deaths = np.array([0, 1, 3, 5])

mean = np.array([mu_alpha, mu_beta])
covariance = np.array([[sigma_alpha ** 2, corr * sigma_alpha * sigma_beta],
                       [corr * sigma_alpha * sigma_beta, sigma_beta**2]])

bioassay_code = """
data {
    int animals[4];
    int deaths[4];
    vector[4] doses;
    vector[2] mean_val;
    matrix[2, 2] covariance;
}

parameters {
    vector[2] alpha_beta;
}

model {
    alpha_beta ~ multi_normal(mean_val, covariance);
    deaths ~ binomial_logit(animals, alpha_beta[1] + alpha_beta[2] * doses);
}
"""

bioassay_dat = {'animals': animals,
                'doses': doses,
                'deaths': deaths,
                'mean_val': mean,
                'covariance': covariance
                }

sm = pystan.StanModel(model_code = bioassay_code)
fit = sm.sampling(data = bioassay_dat, iter=1000, chains=4)

summary = fit.summary()
summary = pd.DataFrame(summary['summary'], columns=summary['summary_colnames'], index=summary['summary_rownames'])
alpha_rhat = summary['Rhat'][0]
beta_rhat = summary['Rhat'][1]

```

```
print('The Rhat values are: {0} {1}'.format(alpha_rhat, beta_rhat))

alpha_beta = fit.extract(permutated=True)['alpha_beta']
alpha = alpha_beta[:,0]
beta = alpha_beta[:,1]

fit.plot()
summary

# Plot the data
plt.scatter(alpha, beta)
plt.xlim(-4, 4)
plt.ylim(-10, 30)
plt.xlabel('Alpha')
plt.ylabel('Beta')
plt.title('Samples')
plt.show()
```

---