

# CS-E5710 Bayesian Data Analysis Assignment

October 26, 2019

## 1 Stan-model

The stan model:

```
1 stan_code = '''
2 data {
3     int<lower=0> n;
4     int<lower=0> deaths[n];
5     int<lower=0> numb_of_animals[n];
6     vector[n] doses;
7     vector[2] mu;
8     cov_matrix[2] cov_m;
9 }
10 parameters {
11     vector[2] alpha_beta;
12 }
13 model {
14     alpha_beta ~ multi_normal(mu, cov_m);
15     deaths ~ binomial_logit(numb_of_animals, alpha_beta[1]
16                             + alpha_beta[2]*doses);
17 }
18 '''
```

Fit the model:

```
1 sm = pystan.StanModel(model_code=stan_code)
2 data = dict(
3     n=len(number_of_animals),
4     deaths=deaths,
5     numb_of_animals=number_of_animals,
6     doses=doses,
7     mu=mean,
8     cov_m=cov_matrix,
9 )
10 fit = sm.sampling(data=data, chains=10, iter=10000, warmup=1000)
```

Outputs:

```

1 fit Inference for Stan model: anon_model_40c9b1c0193fdad43559b9bb79df0201.
2 10 chains, each with iter=10000; warmup=1000; thin=1;
3 post-warmup draws per chain=9000, total post-warmup draws=90000.
4
5           mean se_mean      sd  2.5%   25%   50%   75%  97.5%  n_eff  Rhat
6 alpha_beta[1]  0.97  5.2e-3    0.9  -0.66  0.35  0.93  1.55  2.84 29700  1.0
7 alpha_beta[2] 10.46   0.03   4.57   3.47  7.03  9.87 13.21 20.93 27236  1.0

```

## 2 Rhat value

The Rhat value is  $\hat{R}_\alpha = 1.0$ ,  $\hat{R}_\beta = 1.0$ .

Interpretation:

The potential scale reduction factors is estimated by:

$$\hat{R} = \sqrt{\frac{v\hat{a}r + (\psi|y)}{W}} \quad (1)$$

which declines to 1 as  $n \rightarrow \infty$ . If the potential scale reduction is high, then we have reason to believe that proceeding with further simulations may improve our inference about the target distribution of the associated scalar estimand. If Rhat-value is not close to 1, it is believed that the testing samples may be not from the same distribution or may not converge. In my test all Rhat-values is 1.0 which means the generated chains converge well.

## 3 Scatter plot

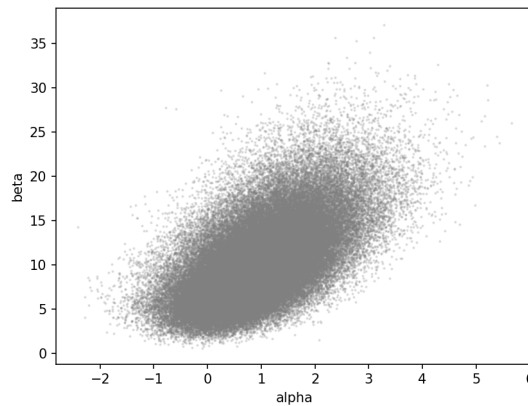


Figure 1: Scatter plot

## A Code

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pystan
4
5 sigma_a = 2
6 sigma_b = 10
7 mu_a = 0
8 mu_b = 10
9 cor = 0.5
10 cov_matrix = np.array([
11     [sigma_a**2,          cor * sigma_a * sigma_b],
12     [cor * sigma_a * sigma_b,  sigma_b**2]
13 ])
14 mean = np.array([mu_a, mu_b])
15
16 doses = np.array([-0.86, -0.3, -0.05, 0.72])
17 deaths = np.array([0, 1, 3, 5])
18 number_of_animals = np.array([5, 5, 5, 5])
19
20 stan_code = '''
21 data {
22     int<lower=0> n;
23     int<lower=0> deaths[n];
24     int<lower=0> numb_of_animals[n];
25     vector[n] doses;
26     vector[2] mu;
27     cov_matrix[2] cov_m;
28 }
29 parameters {
30     vector[2] alpha_beta;
31 }
32 model {
33     alpha_beta ~ multi_normal(mu, cov_m);
34     deaths~binomial_logit(numb_of_animals,alpha_beta[1]
35         +alpha_beta[2]*doses);
36 }
37 '''
38
39 sm = pystan.StanModel(model_code=stan_code)
40 data = dict(
41     n=len(number_of_animals),
42     deaths=deaths,
43     numb_of_animals=number_of_animals,
44     doses=doses,
45     mu=mean,
46     cov_m=cov_matrix,
47 )
48 fit = sm.sampling(data=data, chains=10, iter=10000, warmup=1000)
49 print('fit', fit)
50
51 extracted_samples = fit.extract()
52 samples = extracted_samples['alpha_beta']
53 plt.scatter(samples[:, 0], samples[:, 1], alpha=0.2, s=1, color='grey')
54 plt.ylabel('beta')
```

```
55 plt.xlabel('alpha')
56 plt.savefig('./scatter.png', dpi=150)
```