RStan: Beetles binomial regression example

Heejung Shim

14/10/2019

Install rstan package

```
setwd("/Users/hjshim/Dropbox/MAS2017/lectures/Bayes/Scripts/RStan/")
library(rstan) # load the library
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.19.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

Prepare data for rstan

```
# Load the data
data = read.table(file="beetles-glm-data.txt", header = TRUE)
str(data)
```

```
## 'data.frame':    8 obs. of  3 variables:
##  $ m: int  59 60 62 56 63 59 62 60
##  $ r: int  6 13 18 28 52 53 61 60
##  $ x: num  1.69 1.72 1.76 1.78 1.81 ...
```

```
# Center x
data$centered_x = data$x - mean(data$x)

# Create input data to stan as a list
beetles_data <- list(N = length(data$r), r = data$r, m = data$m, x = data$centered_x)
```

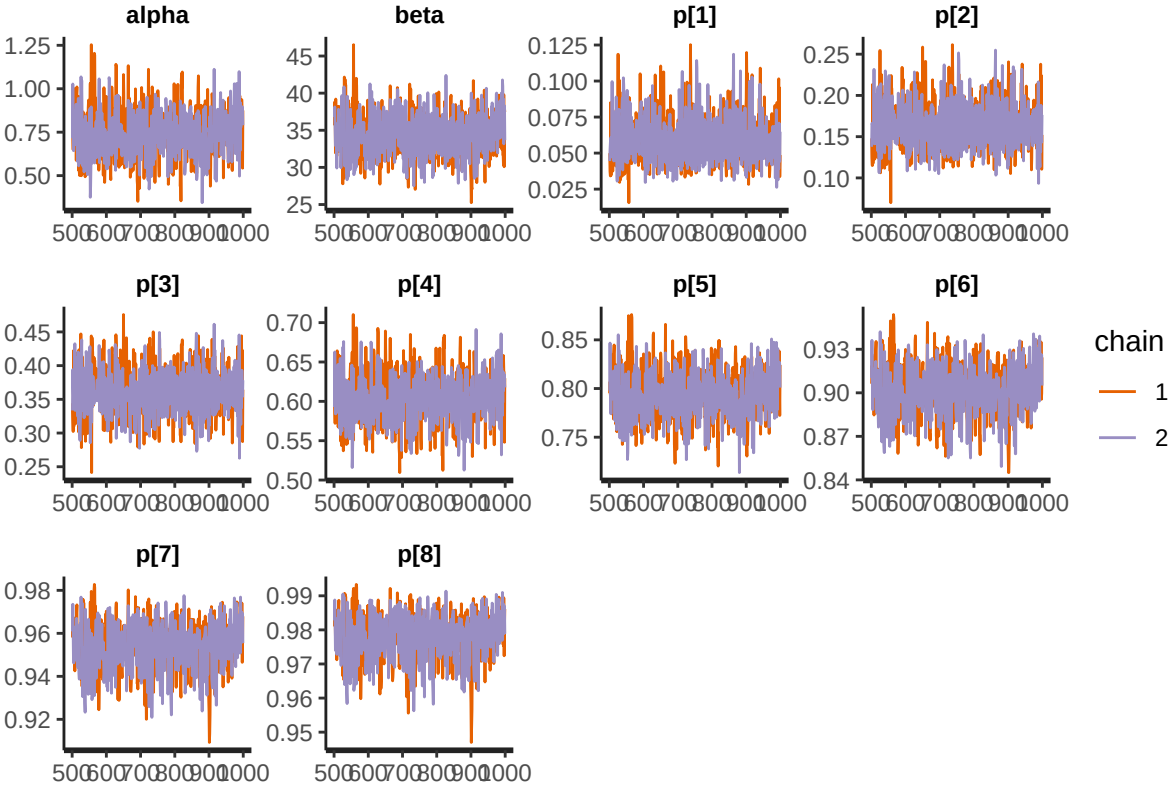
Run rstan and visualise results

```
# run stan
fit <- stan(file = "beetles-glm.stan", data = beetles_data, iter = 1000, chains = 2)
```

```
## hash mismatch so recompiling; make sure Stan code ends with a blank line
```

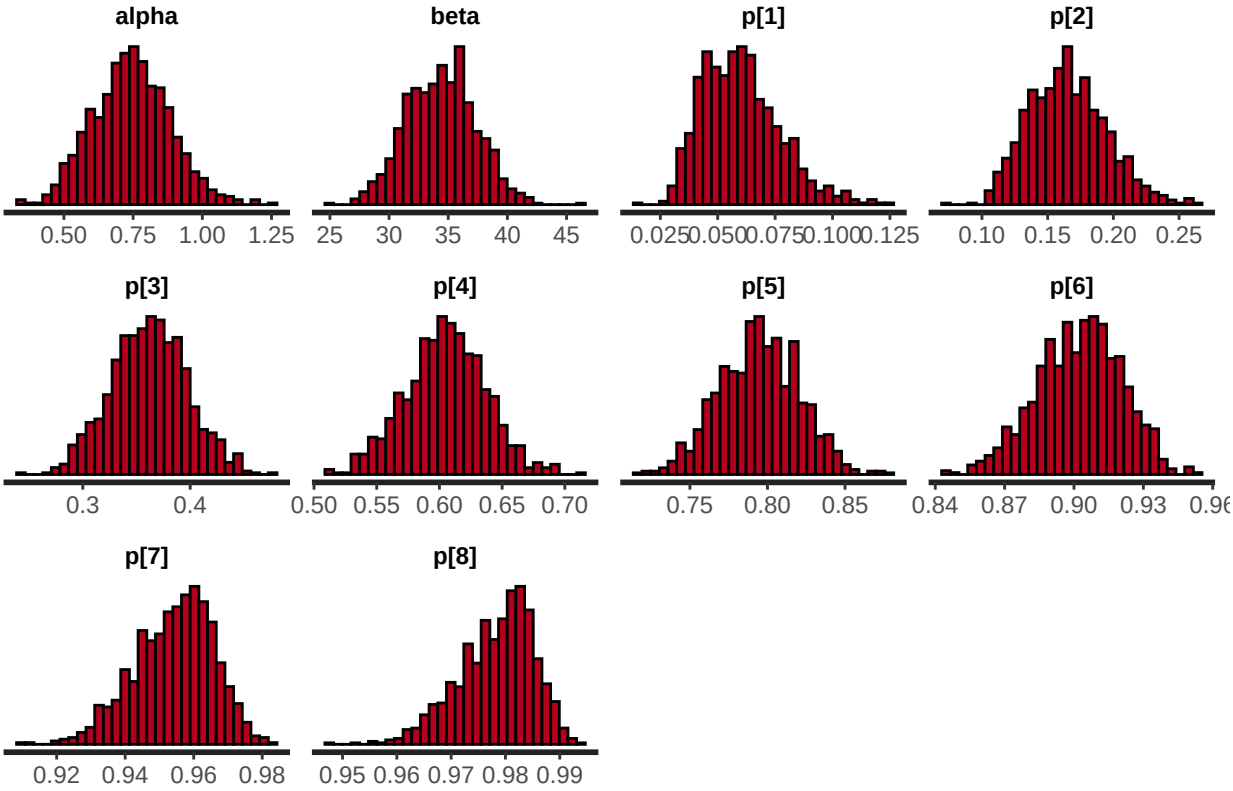
```
# trace plot
plot(fit, plotfun = "trace")
```

```
## 'pars' not specified. Showing first 10 parameters by default.
```

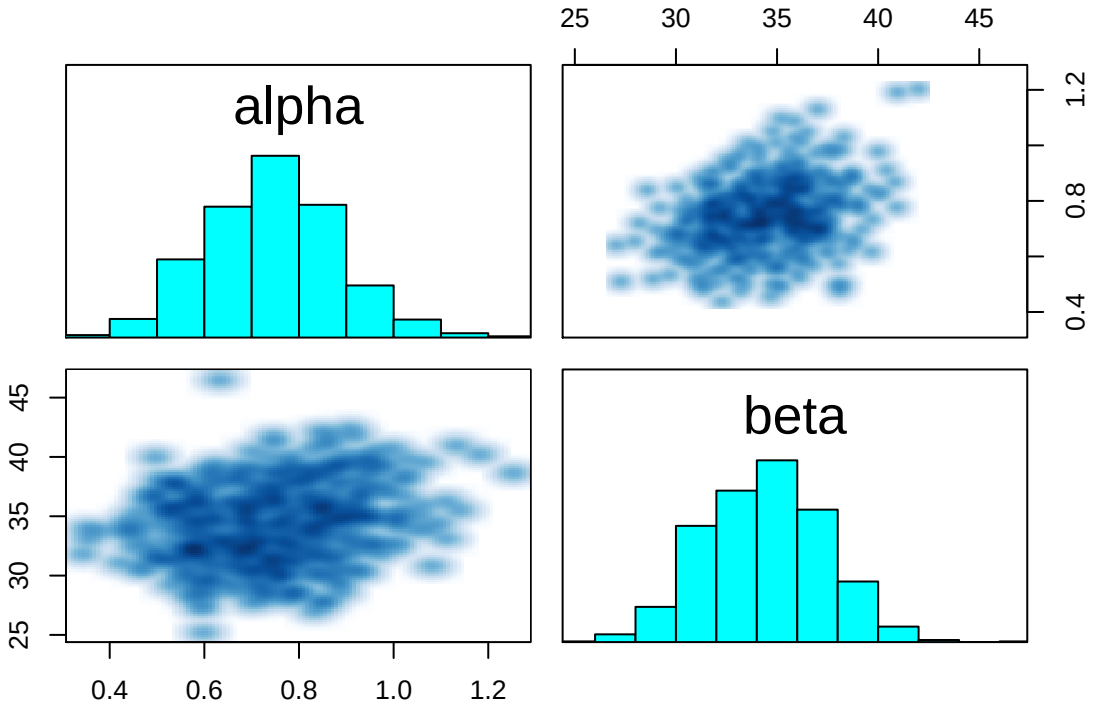



```
# empirical posterior dist
plot(fit, plotfun = "hist")
```

```
## 'pars' not specified. Showing first 10 parameters by default.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



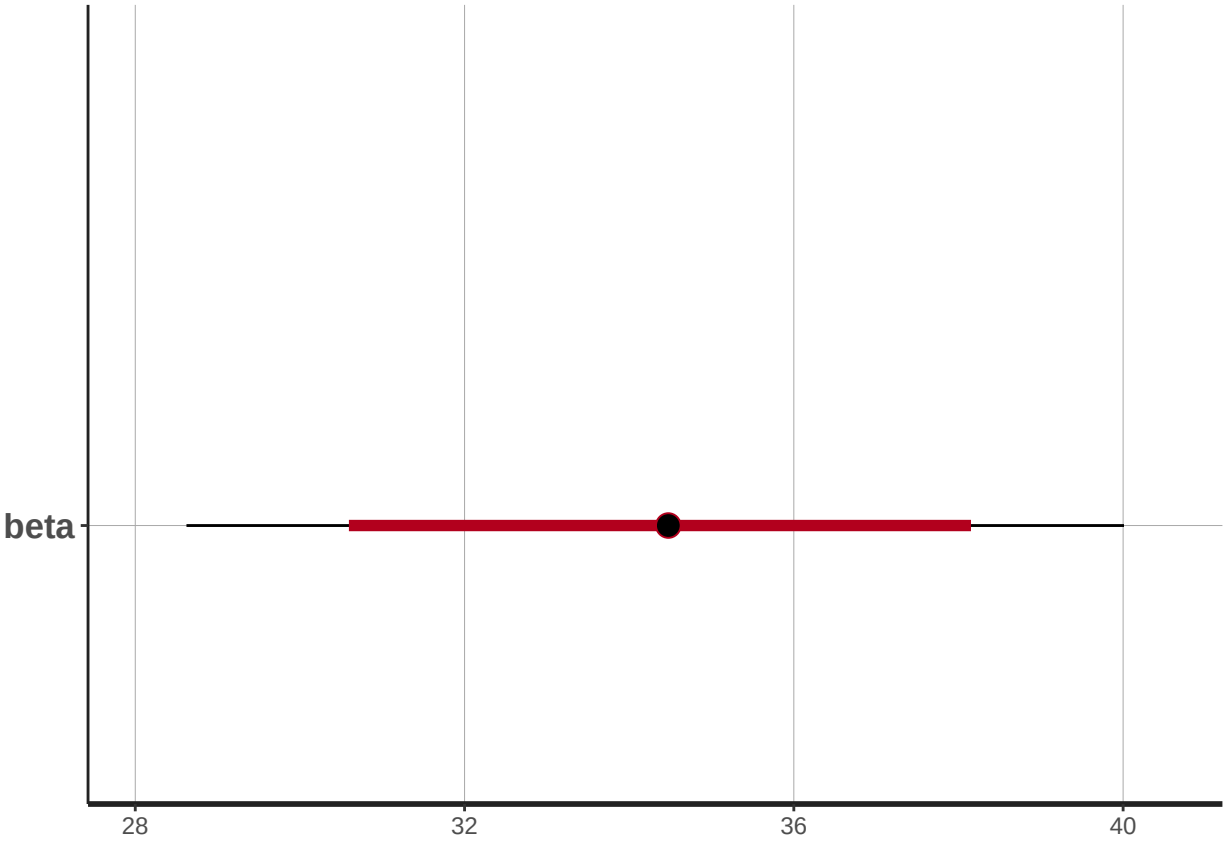

```
# Plot the correlation between the parameters
pairs(fit, pars=c("alpha", "beta"))
```




```
# plotting credible intervals
plot(fit,pars=c("beta"))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```



Want to have an access to summary

```
fit_summary <- summary(fit)
print(fit_summary$summary)
```

```
##                  mean      se_mean          sd          2.5%
## alpha       0.74495659 0.0062846876 0.138611922    0.49314237
## beta       34.40896083 0.1159940279 2.918999552   28.62354972
## p[1]        0.06001147 0.0005566870 0.016794932    0.03372561
## p[2]        0.16496028 0.0008874901 0.029284130    0.11438913
## p[3]        0.36192177 0.0010772791 0.035389565    0.29313032
## p[4]        0.60483180 0.0013684018 0.031939659    0.54053866
## p[5]        0.79454108 0.0012012524 0.025743809    0.74370135
## p[6]        0.90232813 0.0008654634 0.018170752    0.86536822
## p[7]        0.95426649 0.0005417954 0.011494114    0.93084726
## p[8]        0.97828643 0.0003200038 0.006872151    0.96347805
## y_hat[1]    3.54067700 0.0328445337 0.990900961    1.98981095
## y_hat[2]    9.89761667 0.0532494063 1.757047825    6.86334804
## y_hat[3]   22.43914952 0.0667913043 2.194153029   18.17407981
## y_hat[4]   33.87058057 0.0766305013 1.788620903   30.27016476
## y_hat[5]   50.05608814 0.0756789017 1.621859939   46.85318534
## y_hat[6]   53.23735939 0.0510623425 1.072074379   51.05672486
## y_hat[7]   59.16452221 0.0335913119 0.712635045   57.71253004
```

```
## y_hat[8]   58.69718590 0.0192002257 0.412329033   57.80868272
## lp__     -187.31728687 0.0527955135 0.992460356 -189.72007092
##                   25%           50%           75%        97.5%     n_eff
## alpha       0.65420134    0.74393453    0.84106414    1.0253575  486.4450
## beta       32.29548132   34.47533079   36.33520730   40.0105090  633.2815
## p[1]        0.04709533    0.05854816    0.07033073    0.0987449  910.1947
## p[2]        0.14303371    0.16320940    0.18390562    0.2258172 1088.7737
## p[3]        0.33780932    0.36258575    0.38603416    0.4312168 1079.1802
## p[4]        0.58476431    0.60509112    0.62658401    0.6670890  544.7952
## p[5]        0.77608538    0.79499755    0.81312391    0.8431007  459.2795
## p[6]        0.88952693    0.90349235    0.91543598    0.9353314  440.8069
## p[7]        0.94648786    0.95549387    0.96260123    0.9740702  450.0704
## p[8]        0.97376969    0.97934855    0.98330965    0.9893086  461.1850
## y_hat[1]    2.77862429    3.45434155    4.14951315    5.8259490  910.1947
## y_hat[2]    8.58202238    9.79256418   11.03433726   13.5490320 1088.7737
## y_hat[3]   20.94417811   22.48031660   23.93411771   26.7354433 1079.1802
## y_hat[4]   32.74680117   33.88510276   35.08870454   37.3569845  544.7952
## y_hat[5]   48.89337871   50.08484569   51.22680642   53.1153468  459.2795
## y_hat[6]   52.48208881   53.30604893   54.01072265   55.1845511  440.8069
## y_hat[7]   58.68224719   59.24061963   59.68127620   60.3923495  450.0704
## y_hat[8]   58.42618112   58.76091291   58.99857895   59.3585163  461.1850
## lp__     -187.75501690 -187.01186576 -186.60686805 -186.3197349  353.3721
##               Rhat
## alpha    1.0030739
## beta     1.0009133
## p[1]     0.9983511
## p[2]     0.9984214
## p[3]     0.9993658
## p[4]     1.0020382
## p[5]     1.0035861
## p[6]     1.0033755
## p[7]     1.0027985
## p[8]     1.0022519
## y_hat[1] 0.9983511
## y_hat[2] 0.9984214
## y_hat[3] 0.9993658
## y_hat[4] 1.0020382
## y_hat[5] 1.0035861
## y_hat[6] 1.0033755
## y_hat[7] 1.0027985
## y_hat[8] 1.0022519
## lp__     1.0150446
```