RStan: linear model example

Heejung Shim

14/10/2019

Install rstan package

```
setwd("/Users/hjshim/Dropbox/MAS2017/lectures/Bayes/Scripts/RStan/")
#install.packages("rstan", repos = "https://cloud.r-project.org/", dependencies=TRUE)
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)
```

```
#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

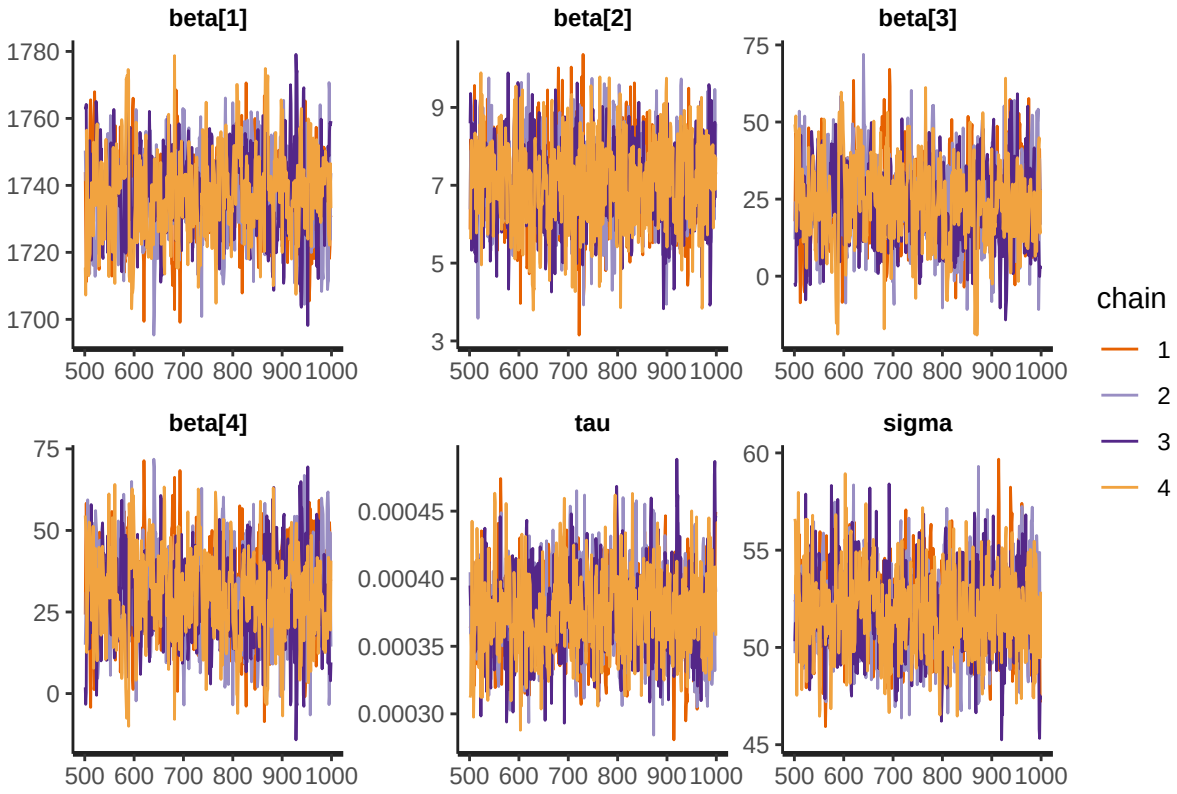
```
# Load the data
data = read.table(file="income-lm-data.txt", header = TRUE)
str(data)
```

```
## 'data.frame':    284 obs. of  3 variables:
##  $ INCOME: num  1673 1743 1747 1735 1749 ...
##  $ AVAGE : num  37.4 39.4 40.6 40.7 41.4 ...
##  $ RURAL : int  2 2 2 2 2 2 2 2 2 2 ...
```

```
# Center the AVAGE varaible
data$centered_AVAGE = data$AVAGE - mean(data$AVAGE)

# Convert the data to a design matrix in R before we pass it to stan
X <- model.matrix(~ data$centered_AVAGE + factor(data$RURAL))

# Create input data to stan as a list
income_data <- list(N = length(data$INCOME), K = ncol(X), X = X, y = data$INCOME)
```

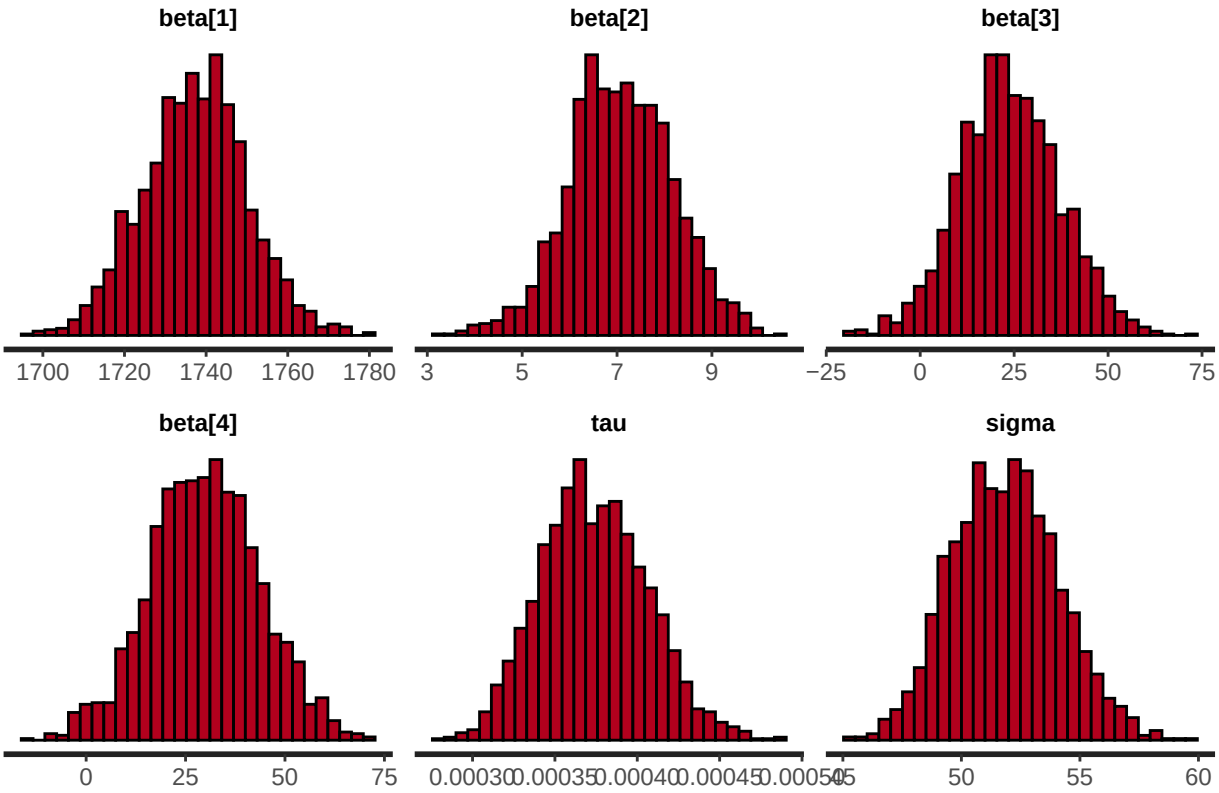
Run rstan and visualise results

```
# run stan
fit <- stan(file = "income-lm.stan", data = income_data, iter = 1000, chains = 4)

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

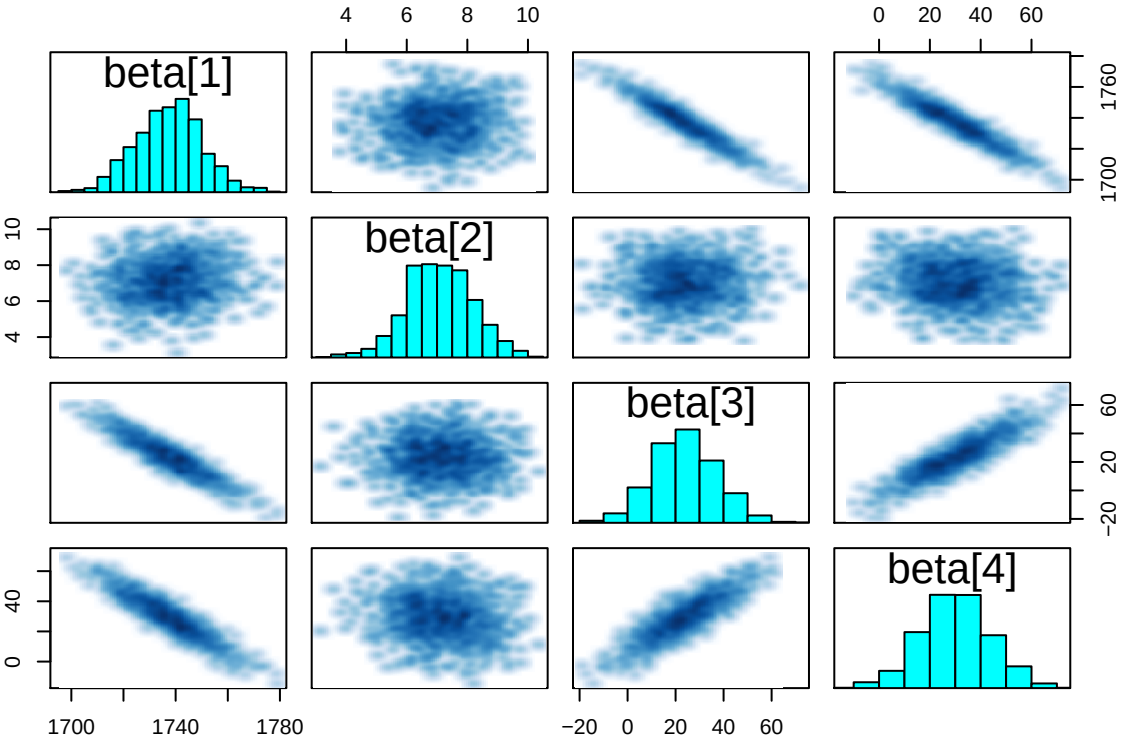



```
# empirical posterior dist
plot(fit, plotfun = "hist")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

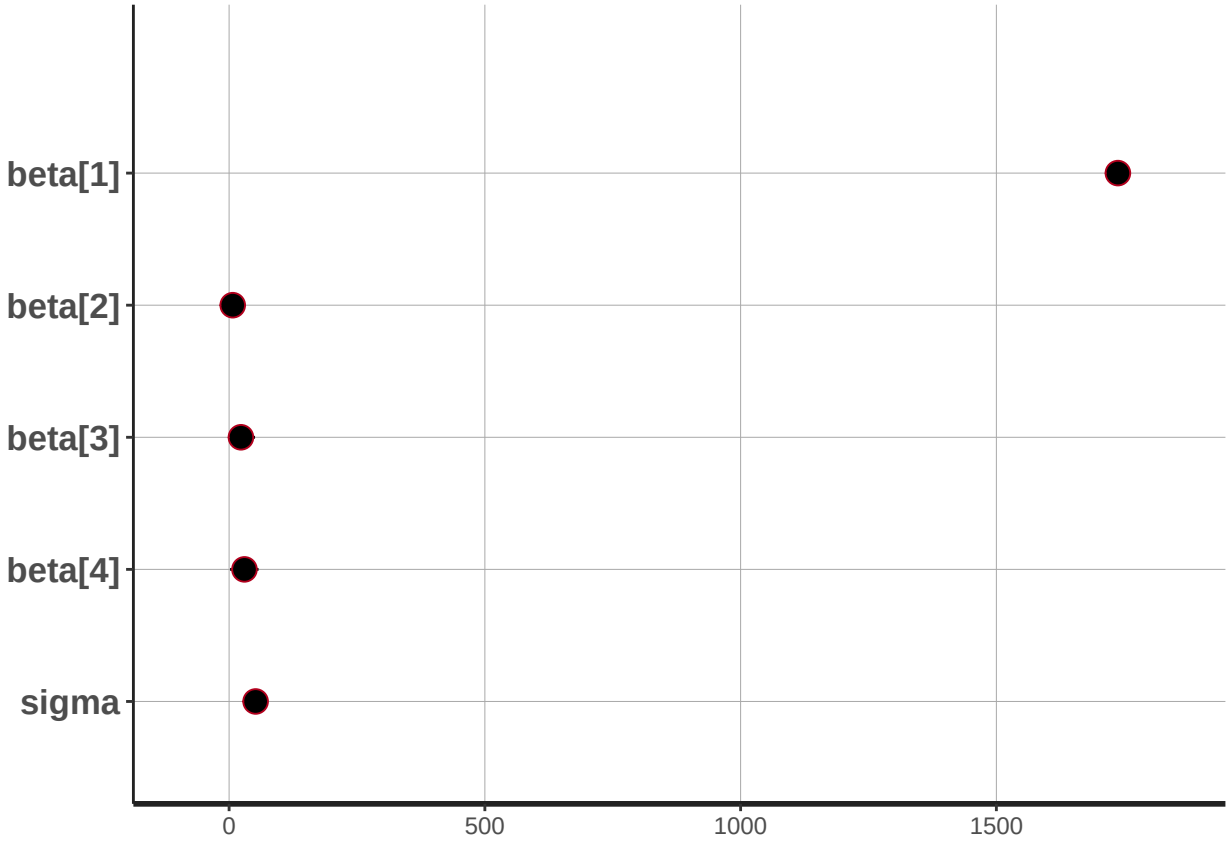



```
# Plot the correlation between the parameters
pairs(fit, pars="beta")
```




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

```
## ci_level: 0.8 (80% intervals)
## outer_level: 0.95 (95% intervals)
```



Want to have an access to samples

```
post_beta<-As.mcmc.list(fit,pars="beta")
length(post_beta)
```

```
## [1] 4
```

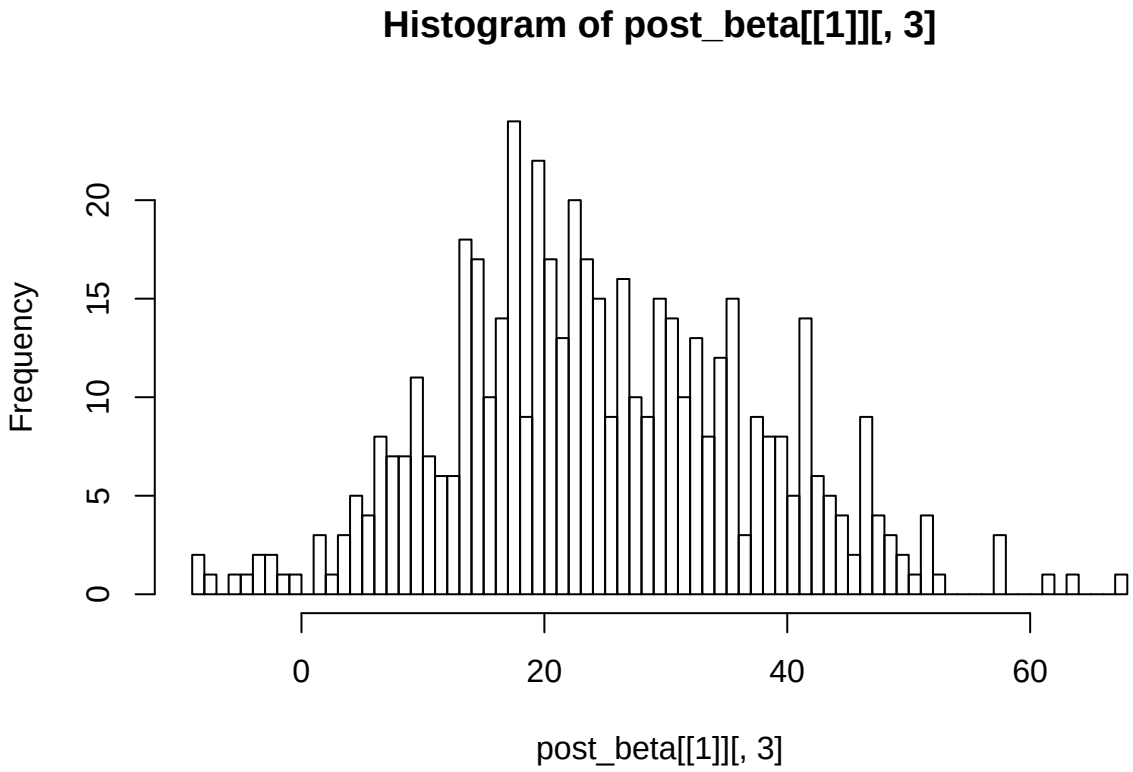
```
str(post_beta[[1]])
```

```
##  'mcmc' num [1:500, 1:4] 1744 1727 1711 1718 1737 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:4] "beta[1]" "beta[2]" "beta[3]" "beta[4]"
##  - attr(*, "mcpar")= num [1:3] 501 1000 1
```

```
dim(post_beta[[1]])
```

```
## [1] 500   4
```

```
hist(post_beta[[1]][,3], breaks = 100)
```


Histogram of post_beta[[1]][, 3]
Frequency
0
5
10
15
20
0
20
40
60
post_beta[[1]][, 3]