```
## smaller font size for chunks
rm(list=ls())
setwd("C:/Users/Ben/Dropbox/Bayesian book/StanCode")
library(rstan)

## Loading required package:  ggplot2
## Loading required package:  StanHeaders
## rstan (Version 2.10.1, packaged:  2016-06-24 13:22:16 UTC, GitRev:
85f7a56811da)
## For execution on a local, multicore CPU with excess RAM we recommend
calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())

library(ggplot2)
options(mc.cores=parallel::detectCores())

Y <- rnorm(10,1.5,0.2)
fit <- stan('StanJags_simpleNormal.stan',iter=200,chains=4,data=list(Y=Y))

## In file included from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/p
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/r
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math.h
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/src/stan/mo
##                  from file11181d3c7077.cpp:8:
## C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/prim/mat/err/check_posi
## C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/prim/mat/err/check_posi
##        typedef typename index_type<Matrix<T_y, Dynamic, 1> >::type size_type;
##                                                                    ^
## In file included from C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/ba
##                  from C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array.hp
##                  from C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/numeric/odeint
##                  from C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/numeric/odeint
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/p
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/p
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/p
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/r
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math.h
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/src/stan/mo
##                  from file11181d3c7077.cpp:8:
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks.hpp: In s
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks.hpp:42:43
##        typedef typename Array::index_range index_range;
##                                           ^
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks.hpp:43:37
##        typedef typename Array::index index;
```

```
##                                         ^
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks
##       typedef typename Array::index_range index_range;
##                                         ^
## C:/Users/Ben/Documents/R/win-library/3.3/BH/include/boost/multi_array/concept_checks
##       typedef typename Array::index index;
##                                         ^
## In file included from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/s
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/s
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/s
##                  from C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/s
##                  from file11181d3c7077.cpp:8:
## C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/core/set_
## C:/Users/Ben/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/core/set_
##     static void set_zero_all_adjoints() {
##                 ^
```

```
print(fit,probs=c(0.25,0.5,0.75))
```

```
## Inference for Stan model: StanJags_simpleNormal.
## 4 chains, each with iter=200; warmup=100; thin=1;
## post-warmup draws per chain=100, total post-warmup draws=400.
##
##                mean se_mean   sd  25%  50%  75% n_eff Rhat
## mu             1.52    0.00 0.08 1.47 1.53 1.56   400 1.00
## sigma          0.34    0.02 0.09 0.28 0.32 0.38    31 1.09
## lSimData[1]    1.51    0.02 0.35 1.32 1.49 1.73   400 0.99
## lSimData[2]    1.56    0.02 0.37 1.32 1.54 1.81   400 0.99
## lSimData[3]    1.53    0.02 0.35 1.31 1.55 1.76   377 1.00
## lSimData[4]    1.53    0.02 0.33 1.31 1.54 1.76   400 1.02
## lSimData[5]    1.52    0.02 0.38 1.27 1.50 1.75   400 0.99
## lSimData[6]    1.52    0.02 0.36 1.30 1.53 1.77   400 1.00
## lSimData[7]    1.50    0.02 0.36 1.27 1.50 1.73   383 1.00
## lSimData[8]    1.54    0.02 0.35 1.33 1.53 1.76   337 0.99
## lSimData[9]    1.49    0.02 0.38 1.29 1.50 1.75   275 1.01
## lSimData[10]   1.51    0.02 0.36 1.30 1.52 1.73   400 1.03
## aMax_indicator 0.50    0.03 0.50 0.00 0.00 1.00   282 1.01
## aMin_indicator 0.40    0.04 0.49 0.00 0.00 1.00   196 1.01
## lp__           4.87    0.09 1.15 4.51 5.18 5.66   167 1.03
##
## Samples were drawn using NUTS(diag_e) at Wed Aug 17 00:31:27 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
##  The estimated Bayesian Fraction of Missing Information is a measure of
```

```
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.9 1.3 1 0.8

mu <- extract(fit,'mu')[[1]]

qplot(mu)

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