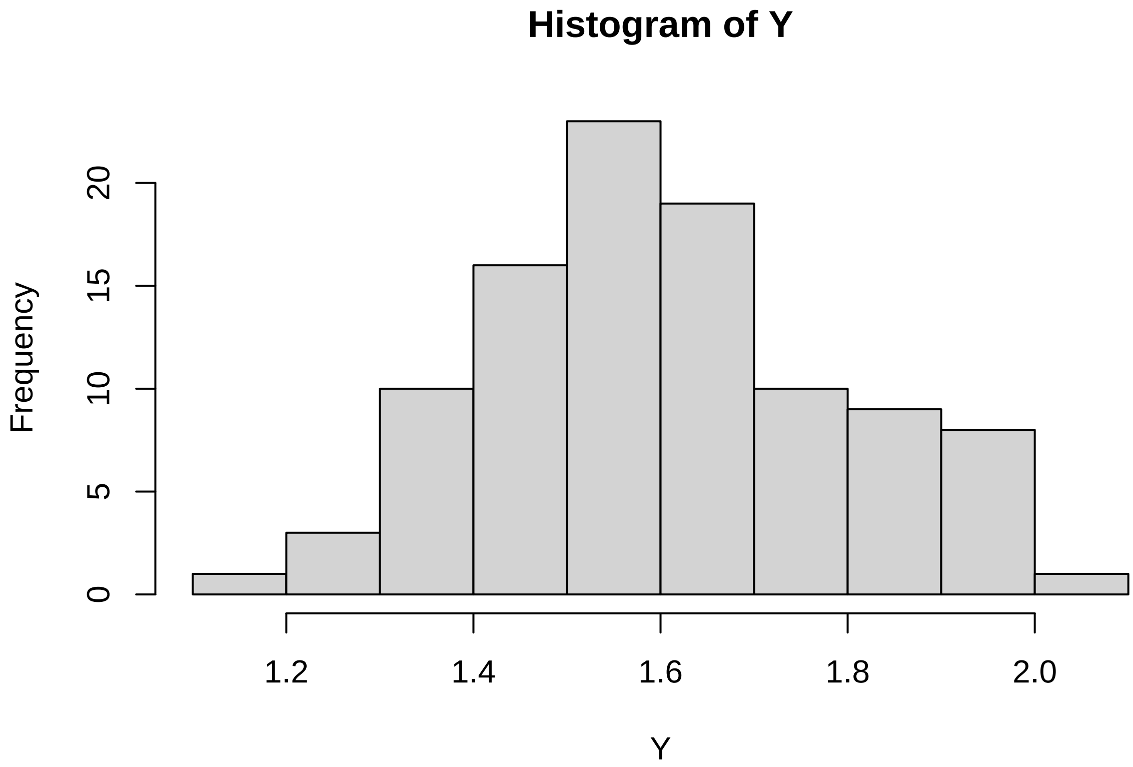
First Model

```
N <- 100
Y <- rnorm(n=N, mean=1.6, sd=0.2)
hist(Y)
```



Compile the model

```
library(rstan)
```

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

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

```
## rstan (Version 2.21.2, 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)
```

```
parallel::detectCores()
```

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

```
model <- stan_model('first_model.stan')
```

```
## Trying to compile a simple C file
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

```
fit <- sampling(model, list(N=N, Y=Y), iter=200, chain=4)
```

```
##
## SAMPLING FOR MODEL 'first_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: WARNING: There aren't enough warmup iterations to fit the
## Chain 1:          three stages of adaptation as currently configured.
## Chain 1:          Reducing each adaptation stage to 15%/75%/10% of
## Chain 1:          the given number of warmup iterations:
## Chain 1:            init_buffer = 15
## Chain 1:            adapt_window = 75
## Chain 1:            term_buffer = 10
## Chain 1:
## Chain 1: Iteration:   1 / 200 [  0%]  (Warmup)
## Chain 1: Iteration:  20 / 200 [ 10%]  (Warmup)
## Chain 1: Iteration:  40 / 200 [ 20%]  (Warmup)
## Chain 1: Iteration:  60 / 200 [ 30%]  (Warmup)
## Chain 1: Iteration:  80 / 200 [ 40%]  (Warmup)
## Chain 1: Iteration: 100 / 200 [ 50%]  (Warmup)
## Chain 1: Iteration: 101 / 200 [ 50%]  (Sampling)
## Chain 1: Iteration: 120 / 200 [ 60%]  (Sampling)
## Chain 1: Iteration: 140 / 200 [ 70%]  (Sampling)
## Chain 1: Iteration: 160 / 200 [ 80%]  (Sampling)
## Chain 1: Iteration: 180 / 200 [ 90%]  (Sampling)
```

```
## Chain 1: Iteration: 200 / 200 [100%]  (Sampling)
## Chain 1: 
## Chain 1:  Elapsed Time: 0.005746 seconds (Warm-up)
## Chain 1:                0.00424 seconds (Sampling)
## Chain 1:                0.009986 seconds (Total)
## Chain 1: 
## 
## SAMPLING FOR MODEL 'first_model' NOW (CHAIN 2).
## Chain 2: 
## Chain 2: Gradient evaluation took 9e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2: 
## Chain 2: 
## Chain 2: WARNING: There aren't enough warmup iterations to fit the
## Chain 2:          three stages of adaptation as currently configured.
## Chain 2:          Reducing each adaptation stage to 15%/75%/10% of
## Chain 2:          the given number of warmup iterations:
## Chain 2:            init_buffer = 15
## Chain 2:            adapt_window = 75
## Chain 2:            term_buffer = 10
## Chain 2: 
## Chain 2: Iteration:   1 / 200 [  0%]  (Warmup)
## Chain 2: Iteration:  20 / 200 [ 10%]  (Warmup)
## Chain 2: Iteration:  40 / 200 [ 20%]  (Warmup)
## Chain 2: Iteration:  60 / 200 [ 30%]  (Warmup)
## Chain 2: Iteration:  80 / 200 [ 40%]  (Warmup)
## Chain 2: Iteration: 100 / 200 [ 50%]  (Warmup)
## Chain 2: Iteration: 101 / 200 [ 50%]  (Sampling)
## Chain 2: Iteration: 120 / 200 [ 60%]  (Sampling)
## Chain 2: Iteration: 140 / 200 [ 70%]  (Sampling)
## Chain 2: Iteration: 160 / 200 [ 80%]  (Sampling)
## Chain 2: Iteration: 180 / 200 [ 90%]  (Sampling)
## Chain 2: Iteration: 200 / 200 [100%]  (Sampling)
## Chain 2: 
## Chain 2:  Elapsed Time: 0.005764 seconds (Warm-up)
## Chain 2:                0.00458 seconds (Sampling)
## Chain 2:                0.010344 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'first_model' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 1e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.1 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: WARNING: There aren't enough warmup iterations to fit the
## Chain 3:          three stages of adaptation as currently configured.
## Chain 3:          Reducing each adaptation stage to 15%/75%/10% of
## Chain 3:          the given number of warmup iterations:
## Chain 3:            init_buffer = 15
## Chain 3:            adapt_window = 75
## Chain 3:            term_buffer = 10
## Chain 3:
## Chain 3: Iteration:   1 / 200 [  0%]  (Warmup)
## Chain 3: Iteration:  20 / 200 [ 10%]  (Warmup)
## Chain 3: Iteration:  40 / 200 [ 20%]  (Warmup)
## Chain 3: Iteration:  60 / 200 [ 30%]  (Warmup)
## Chain 3: Iteration:  80 / 200 [ 40%]  (Warmup)
## Chain 3: Iteration: 100 / 200 [ 50%]  (Warmup)
## Chain 3: Iteration: 101 / 200 [ 50%]  (Sampling)
## Chain 3: Iteration: 120 / 200 [ 60%]  (Sampling)
## Chain 3: Iteration: 140 / 200 [ 70%]  (Sampling)
## Chain 3: Iteration: 160 / 200 [ 80%]  (Sampling)
## Chain 3: Iteration: 180 / 200 [ 90%]  (Sampling)
## Chain 3: Iteration: 200 / 200 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.006069 seconds (Warm-up)
## Chain 3:                0.004591 seconds (Sampling)
## Chain 3:                0.01066 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'first_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: WARNING: There aren't enough warmup iterations to fit the
## Chain 4:          three stages of adaptation as currently configured.
## Chain 4:          Reducing each adaptation stage to 15%/75%/10% of
## Chain 4:          the given number of warmup iterations:
## Chain 4:            init_buffer = 15
## Chain 4:            adapt_window = 75
## Chain 4:            term_buffer = 10
## Chain 4:
## Chain 4: Iteration:   1 / 200 [  0%]  (Warmup)
## Chain 4: Iteration:  20 / 200 [ 10%]  (Warmup)
## Chain 4: Iteration:  40 / 200 [ 20%]  (Warmup)
## Chain 4: Iteration:  60 / 200 [ 30%]  (Warmup)
## Chain 4: Iteration:  80 / 200 [ 40%]  (Warmup)
## Chain 4: Iteration: 100 / 200 [ 50%]  (Warmup)
## Chain 4: Iteration: 101 / 200 [ 50%]  (Sampling)
## Chain 4: Iteration: 120 / 200 [ 60%]  (Sampling)
## Chain 4: Iteration: 140 / 200 [ 70%]  (Sampling)
## Chain 4: Iteration: 160 / 200 [ 80%]  (Sampling)
## Chain 4: Iteration: 180 / 200 [ 90%]  (Sampling)
## Chain 4: Iteration: 200 / 200 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.004642 seconds (Warm-up)
## Chain 4:                0.004921 seconds (Sampling)
## Chain 4:                0.009563 seconds (Total)
## Chain 4:
```

```
## Warning: The largest R-hat is 1.05, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

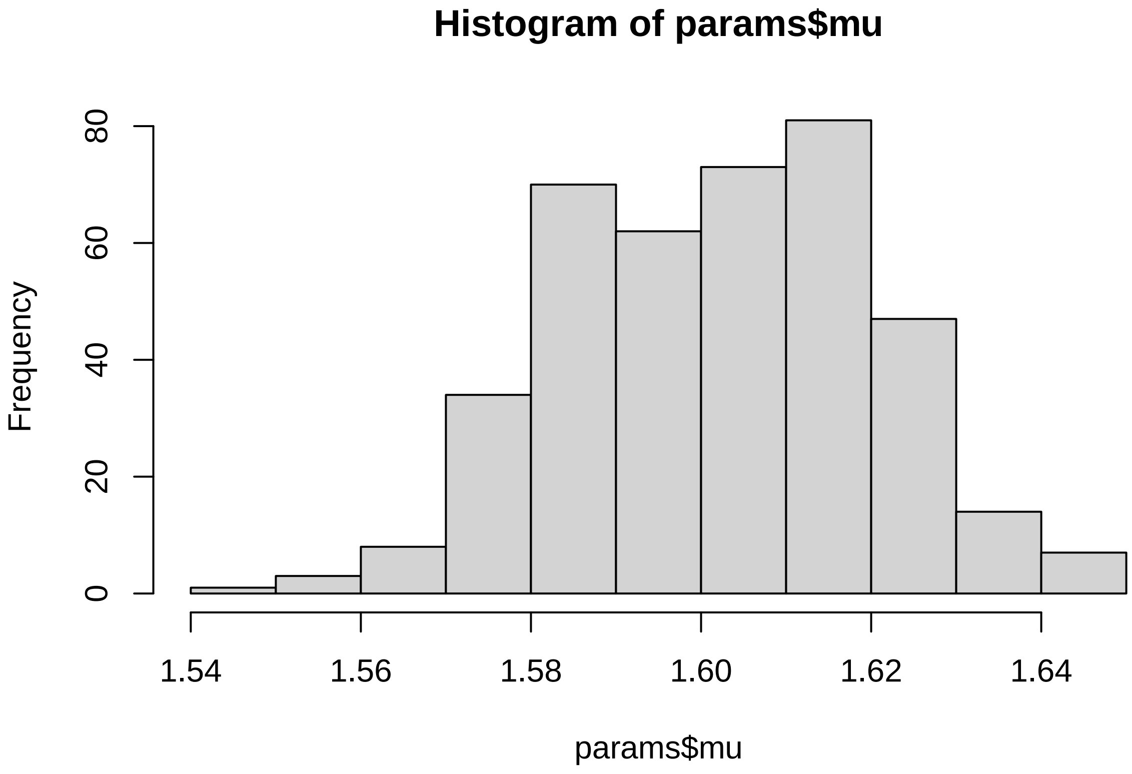
```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
print(fit)
```

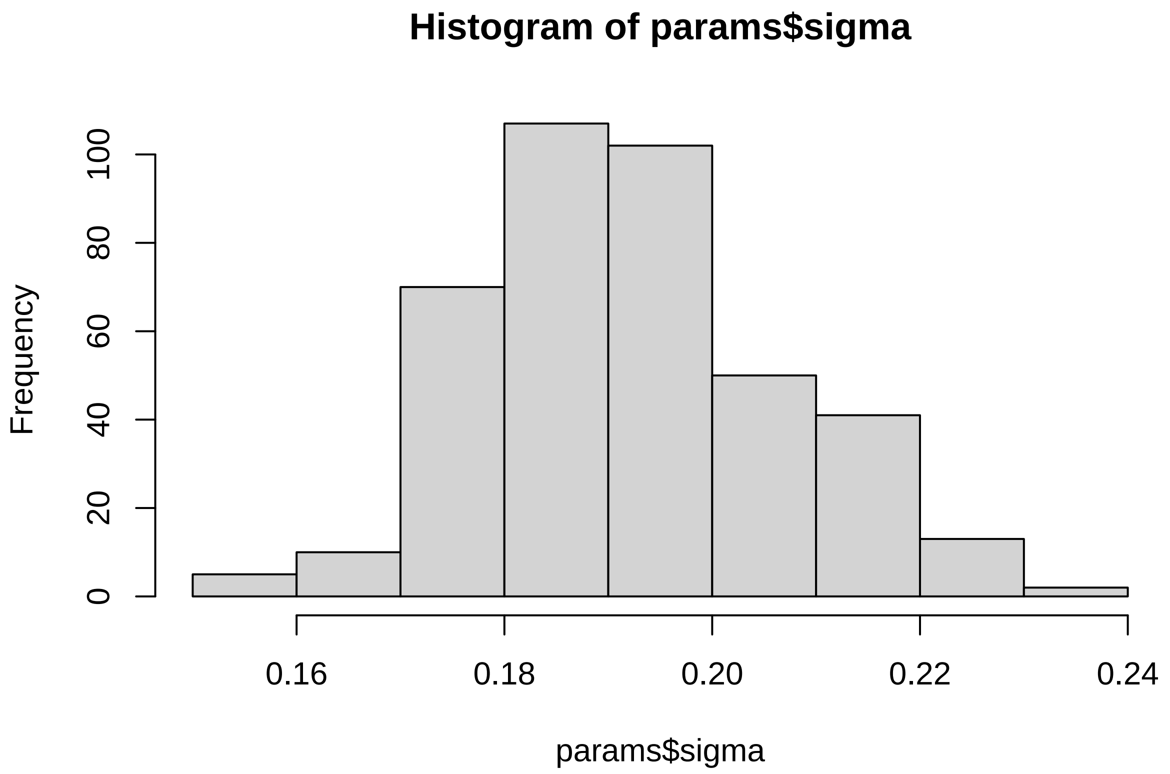
```
## Inference for Stan model: first_model.
## 4 chains, each with iter=200; warmup=100; thin=1;
## post-warmup draws per chain=100, total post-warmup draws=400.
##
##         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## mu      1.60    0.00 0.02   1.57   1.59   1.60   1.62   1.64   359 1.01
## sigma   0.19    0.00 0.01   0.17   0.18   0.19   0.20   0.22   134 1.06
## lp__  114.54    0.09 0.93 111.78 114.16 114.80 115.23 115.51   118 1.02
##
## Samples were drawn using NUTS(diag_e) at Tue Aug 25 12:45:14 2020.
## 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).
```

```
params <- extract(fit)
```

```
hist(params$mu)
```



```
hist(params$sigma)
```



```
library(shinystan)
```

```
## Loading required package: shiny

##
## This is shinystan version 2.5.0
```

```
launch_shinystan(fit)
```

```
##
## Launching ShinyStan interface... for large models this  may take some time.

##
## Listening on http://127.0.0.1:4604
```