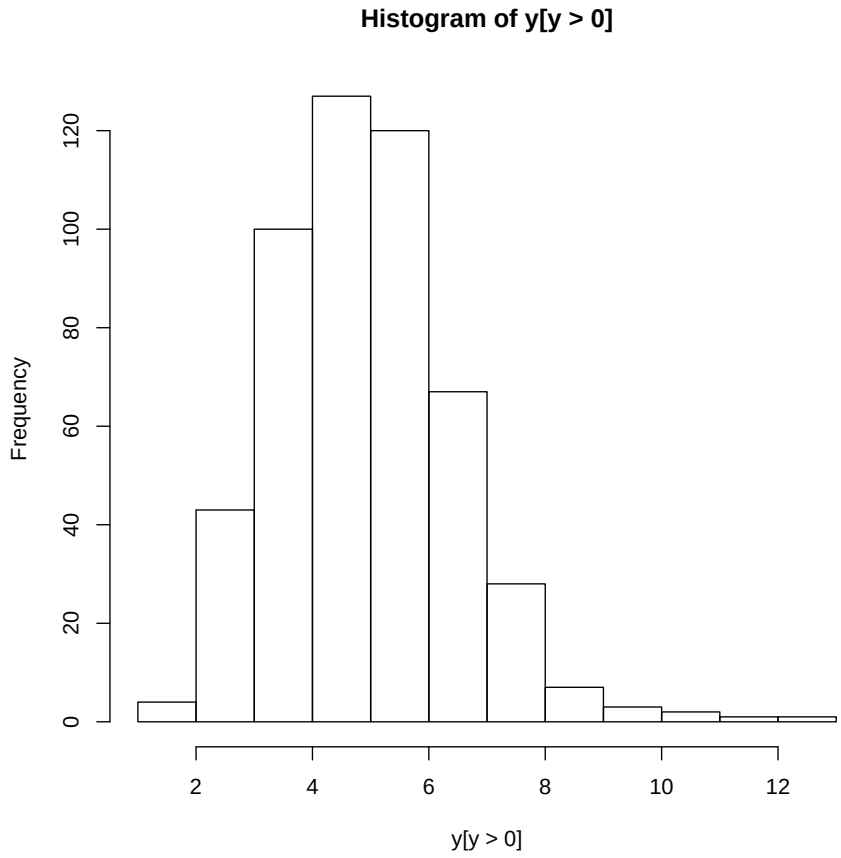
```r
theta<-0.27      #probability of rain on a given day
alpha=10          #shape parameter for gamma
beta=2           #scale parameter for gamma
years<-5
N<-365*years     #sample size

rained<-rbinom(N,1,theta)        #generate binary rain/no rain vector
rainfall<-rgamma(N,alpha,beta)   #generate rainfall quantity vector

y<-rained*rainfall               #observed values are product of these

y[1:100]                         #print the first few y values
```

```
##   [1] 0.000000 5.338024 0.000000 0.000000 4.967348 0.000000 2.949090
##   [8] 0.000000 6.557686 4.527136 0.000000 0.000000 0.000000 5.335668
##  [15] 0.000000 0.000000 5.872335 4.318328 0.000000 0.000000 0.000000
##  [22] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.989063
##  [29] 0.000000 4.751420 4.109016 0.000000 0.000000 0.000000 4.421470
##  [36] 6.303297 0.000000 0.000000 5.494433 6.225622 0.000000 0.000000
##  [43] 0.000000 0.000000 4.355905 0.000000 0.000000 0.000000 0.000000
##  [50] 0.000000 6.105776 3.541700 0.000000 0.000000 0.000000 0.000000
##  [57] 0.000000 6.575496 0.000000 0.000000 0.000000 0.000000 4.406687
##  [64] 0.000000 3.358243 0.000000 0.000000 0.000000 0.000000 0.000000
##  [71] 4.494552 0.000000 0.000000 0.000000 0.000000 0.000000 5.154375
##  [78] 3.182705 4.454077 8.042181 1.042064 5.309208 0.000000 0.000000
##  [85] 4.498863 0.000000 6.056975 0.000000 6.116791 0.000000 0.000000
##  [92] 0.000000 4.413903 0.000000 0.000000 2.659249 3.153756 5.323131
##  [99] 0.000000 0.000000
```

```r
hist(y[y>0])                     #histogram of measurable rainfall amounts
```



Call STAN

```
library(rstan)

## Loading required package:  ggplot2
## rstan (Version 2.9.0-3, packaged:  2016-02-11 15:54:41 UTC, GitRev:
05c3d0058b6a)
## For execution on a local, multicore CPU with excess RAM we recommend
calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())

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

stanfit<-stan("gamma_test.stan") #call STAN
print(stanfit)                   #print a summary of the results
```

```
## Inference for Stan model: gamma_test.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd     2.5%      25%      50%      75%    97.5%
## theta     0.28    0.00 0.01     0.26     0.27     0.28     0.28     0.30
## alpha    10.30    0.02 0.62     9.15     9.87    10.29    10.71    11.53
## beta      2.10    0.00 0.13     1.86     2.01     2.10     2.19     2.36
## lp__  -1983.52    0.04 1.23 -1986.73 -1984.06 -1983.20 -1982.64 -1982.13
##       n_eff Rhat
## theta  1396 1.00
## alpha   715 1.01
## beta    709 1.01
## lp__   1039 1.00
##
## Samples were drawn using NUTS(diag_e) at Sat May 21 05:01:56 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).
```

Launch shinystan

```
library(shinystan)              #downloaded from cran

## Loading required package:  shiny
##
## This is shinystan version 2.1.0

launch_shinystan(stanfit)

##
## Loading...
## Note:  for large models ShinyStan may take a few moments to launch.
##
## Listening on http://127.0.0.1:4872
```