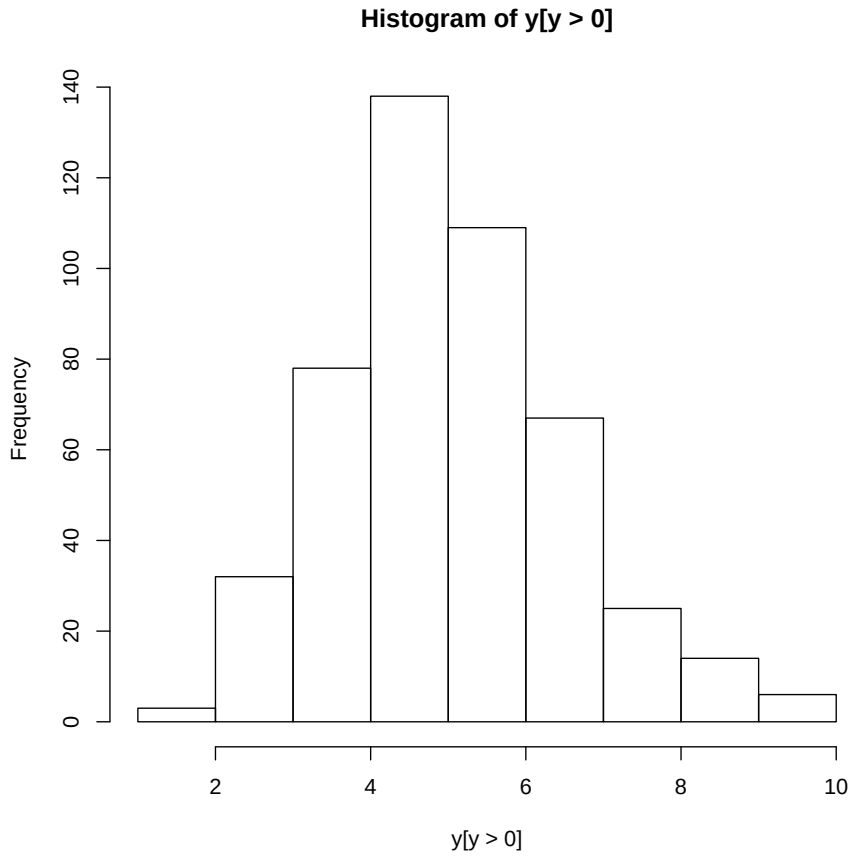
```r
theta<-0.27       #probability of rain on a given day
alpha=10          #shape parameter for gamma
beta=2            #scale parameter for gamma
years<-5
N<-365*years      #sample size

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] 4.732445 0.000000 6.581730 4.862636 0.000000 0.000000 3.782456
##   [8] 4.142779 0.000000 0.000000 5.053181 0.000000 0.000000 0.000000
##  [15] 0.000000 0.000000 0.000000 0.000000 0.000000 4.370112 4.102435
##  [22] 0.000000 0.000000 0.000000 0.000000 1.895060 0.000000 6.914596
##  [29] 5.138466 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
##  [36] 0.000000 0.000000 0.000000 0.000000 0.000000 6.567319 0.000000
##  [43] 4.016615 0.000000 4.181984 3.166896 0.000000 0.000000 0.000000
##  [50] 3.436176 0.000000 0.000000 0.000000 5.184088 0.000000 0.000000
##  [57] 5.896333 0.000000 6.677558 0.000000 0.000000 0.000000 0.000000
##  [64] 0.000000 0.000000 0.000000 5.451706 0.000000 0.000000 5.662272
##  [71] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
##  [78] 0.000000 4.479577 0.000000 0.000000 0.000000 0.000000 0.000000
##  [85] 0.000000 0.000000 0.000000 0.000000 5.212412 0.000000 0.000000
##  [92] 0.000000 0.000000 0.000000 0.000000 0.000000 4.147014 6.700813
##  [99] 0.000000 4.351337
```

```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     0.26    0.00 0.01     0.24     0.25     0.26     0.27     0.28
## alpha    10.93    0.02 0.69     9.65    10.46    10.91    11.40    12.28
## beta      2.18    0.00 0.14     1.91     2.08     2.17     2.27     2.45
## lp__  -1894.62    0.04 1.19 -1897.68 -1895.18 -1894.33 -1893.73 -1893.22
##       n_eff Rhat
## theta  1035    1
## alpha  1110    1
## beta   1115    1
## lp__    980    1
##
## Samples were drawn using NUTS(diag_e) at Sat May 21 05:35:23 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).
```

```
print(get_stanmodel(stanfit))    #print the model file
```

```
## S4 class stanmodel 'gamma_test' coded as follows:
## //mixture model
## //  probability of measurable rain is Bernoulli with parameter theta
## //  rainfall amount given measurable rainfall is gamma(alpha,beta)
## //
## data {
##   int<lower=1> N;               //number of observations
##   real<lower=0> y[N];           //rainfall amount
## }
## parameters {
##   real<lower=0,upper=1> theta;  //probability of measurable rain
##   real<lower=0> alpha;          //shape parameter for rainfall amount gamma
##   real<lower=0> beta;           //scale parameter for rainfall amount gamma
## }
## model {
##   theta ~ beta(1,1);            //uniform prior for theta
##   alpha ~ normal(0,5);          //half-normal prior for alpha
##   beta ~ normal(0,5);           //half-normal prior for beta
##
##   for (i in 1:N){
##     if(y[i] < 0.0001) {                  //no measurable rain
##       increment_log_prob(log(1-theta));
##     }
##     else {                               //measurable rainfall
```

```
##       increment_log_prob(log(theta)+gamma_log(y[i],alpha,beta));
##     }
##   }
## }
```

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://127.0.0.1:5391
```