Simulation Study for Timing Model

James Watson

07/05/2019

Global options for stan model runs

```
Chains = 4
options(mc.cores = Chains)
IT = 10^4
WarmUp = .5*IT
thin = 40
Sim_Iterations = 50
```

Global simulation sample sizes and follow-up time

```
N_PMQ = 800
N_AS = 200
N_CQ = 200
FUP_time = 300
```

Data simulation functions

We explore a few scenarios of increasing complexity to check:

- The model can recover correct parameters when the data generating process is correctly specified (sanity check)
- The effect of a mis-specified data generating process. For this we look at the impact of seasonality

Data simulation for model 1

This simulates data under the assumptions of Model 1.

We set the simulation parameters:

```
params_M1 = list(lambda = 1/1200,
                 gamma = 1/80,
                 lambda_recrud = 1/10,
                 logit_EarlyL = 0.5,
                 logit_c1_CQ = 0.01,
                 logit_c1_AS = 0.01,
                 logit_mean_p = logit(0.2),
                 logit_sd_p = (-logit(0.2)+logit(0.8))/1.96,
                 AS_shape = 10,
                 AS_scale = 21,
                 CQ_shape = 10,
                 CQ_scale = 42,
                 rate_decrease = .66)
```

Generate data with these parameters:

```
set.seed(475732)
out1 = simulate_dataset(N_PMQ = N_PMQ, N_CQ = N_CQ,
                        N_AS = N_AS,FUP_time = FUP_time,
                        data_generation_function = generate_patient_data_Model1,
                        params = params_M1,
                        Study_Period = c(rep(1,N_PMQ/2),rep(2,N_PMQ/2 + N_AS + N_CQ)))
Simdata_Model1 = out1$Simdata
Simulation_truth1 = out1$Simulation_truth
```

Run Stan Model 1 on simulated data generated from Model 1

Load or compile the stan model 1

```
if(RECOMPILE_MODELS){
  writeLines('Compiling model 1....')
  source('../Timing_Model/StanModel1.R')
  save(Timing_Model1, file = '../RData/TimingModel/Timing_Model1.RData')
} else {
  load('../RData/TimingModel/Timing_Model1.RData')
}
```

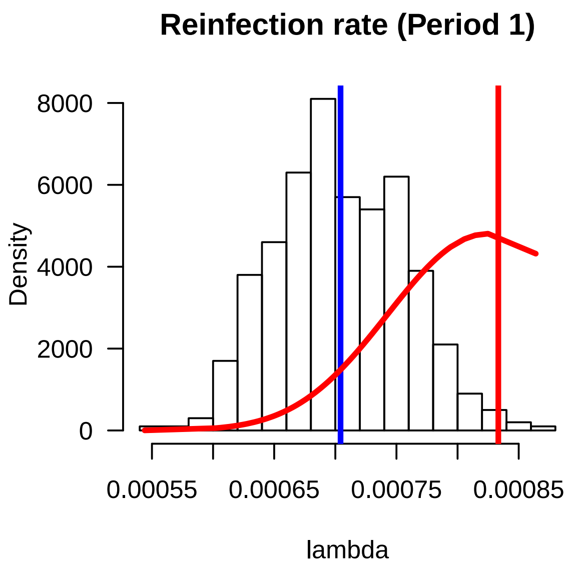
Prior specification

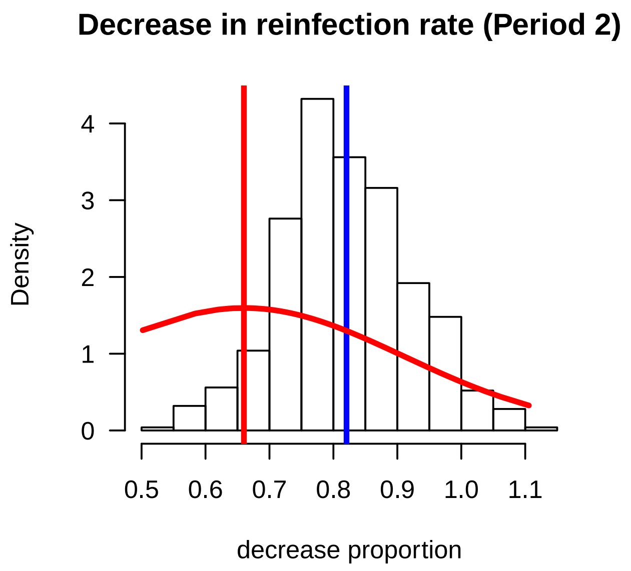
```
# The hierachical parameters defining the prior distributions for model 1
Prior_params_M1 = list(Hyper_lambda_shape = 100,
                       Hyper_lambda_rate = 100*(1/params_M1$lambda),
                       Hyper_gamma_shape = 100,
                       Hyper_gamma_rate = 100*(1/params_M1$gamma),
                       Hyper_lambda_recrud_shape = 100,
                       Hyper_lambda_recrud_rate = 100*(1/params_M1$lambda_recrud),
                       Hyper_AS_shape_mean = params_M1$AS_shape,
                       Hyper_AS_shape_sd = 1,
                       Hyper_AS_scale_mean = params_M1$AS_scale,
                       Hyper_AS_scale_sd = 1,
                       Hyper_CQ_shape_mean = params_M1$CQ_shape,
                       Hyper_CQ_shape_sd = 1,
                       Hyper_CQ_scale_mean = params_M1$CQ_scale,
                       Hyper_CQ_scale_sd = 2,
                       Hyper_logit_mean_p_mean = params_M1$logit_mean_p,
                       Hyper_logit_mean_p_sd = 1,
                       Hyper_logit_sd_p_lambda = 1,
                       Hyper_logit_c1_mean = params_M1$logit_c1_AS,
                       Hyper_logit_c1_sd = .25,
                       Early_L_logit_mean = params_M1$logit_EarlyL,
                       Early_L_logit_sd = .5,
                       Hyper_mean_rate_decrease = params_M1$rate_decrease,
                       Hyper_sd_rate_decrease = 0.25)
```

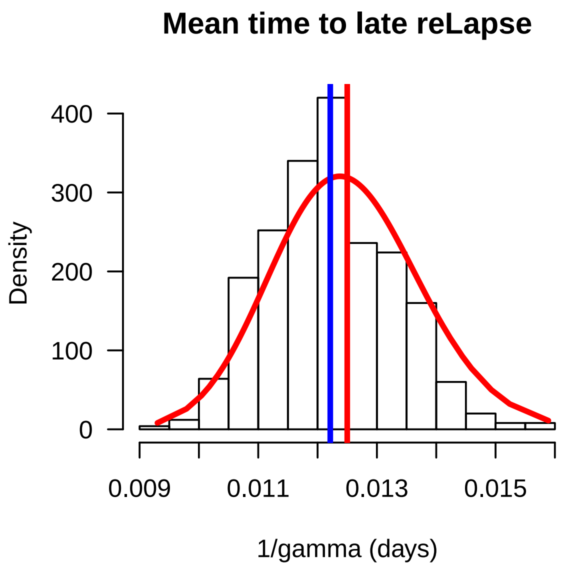
Fit stan model to simulated data

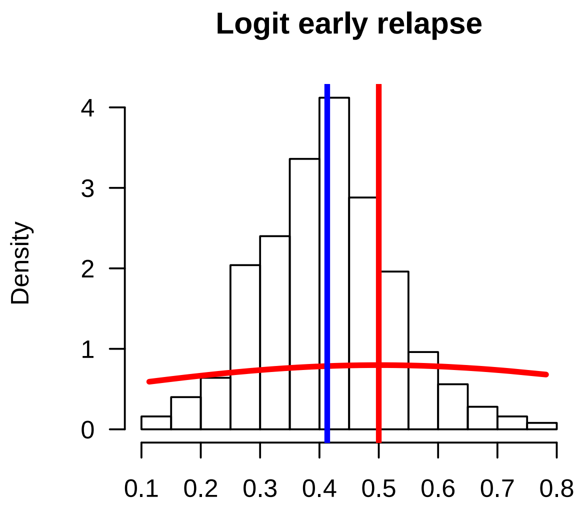
Plot output, comparing prior, ground truth and model estimate. This gives a qualitative assessement but for a single simulation run.

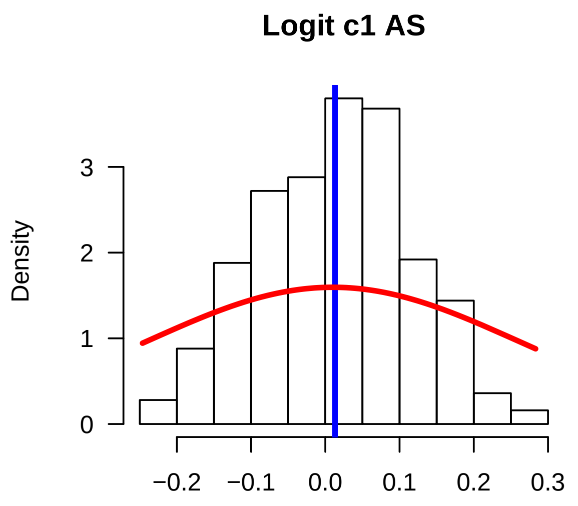
```
plot_output_model1(thetas_mod1,Simulation_truth1,
                   Simdata_Model1,Prior_params_M1)
```

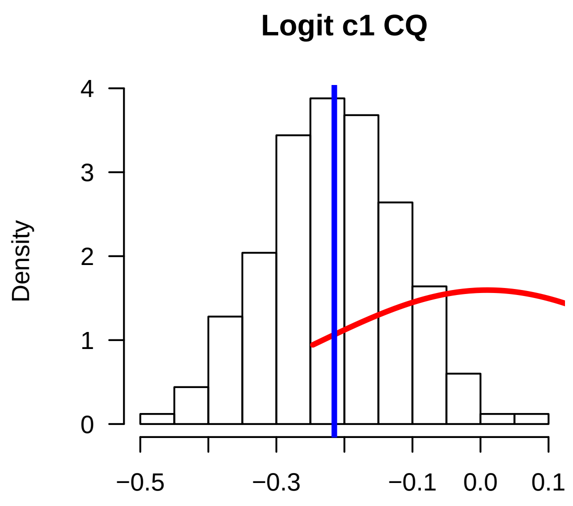

Reinfection rate (Period 1)
Density
8000
6000
4000
2000
0
0.00055
0.00065
0.00075
0.00085
lambda

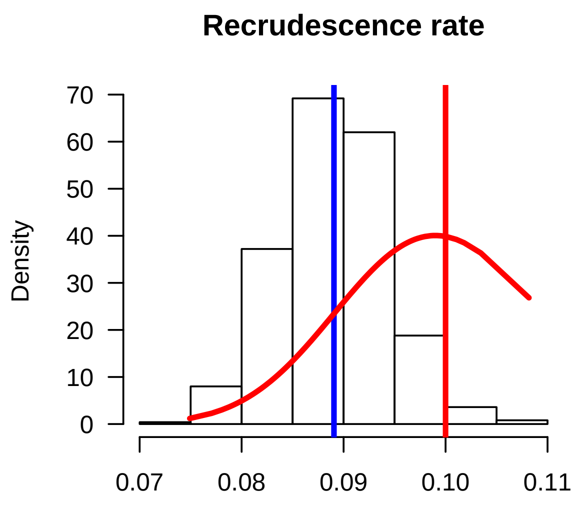

Decrease in reinfection rate (Period 2)
Density
4
3
2
1
0
0.5
0.6
0.7
0.8
0.9
1.0
1.1
decrease proportion

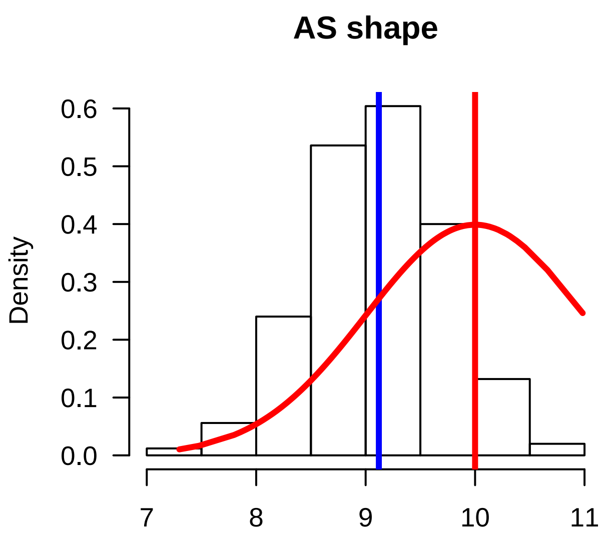

Mean time to late reLapse
Density
400
300
200
100
0
0.009
0.011
0.013
0.015
1/gamma (days)

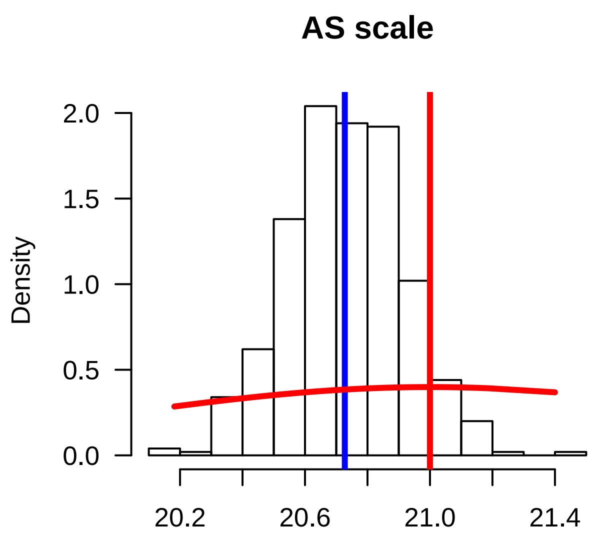

Logit early relapse
Density
4
3
2
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

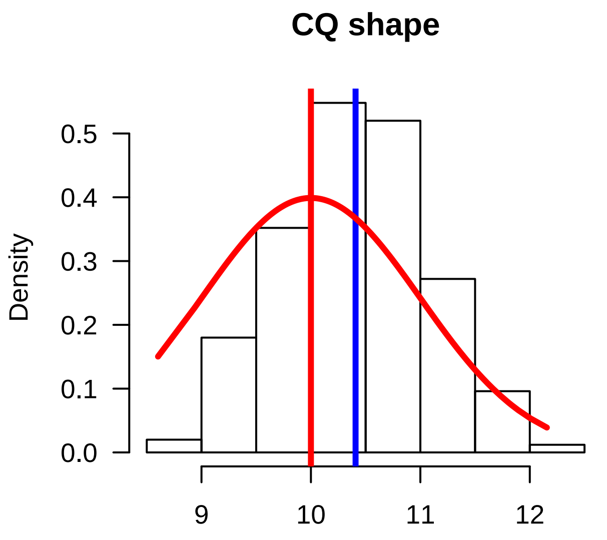

Logit c1 AS
Density
0
1
2
3
−0.2
−0.1
0.0
0.1
0.2
0.3

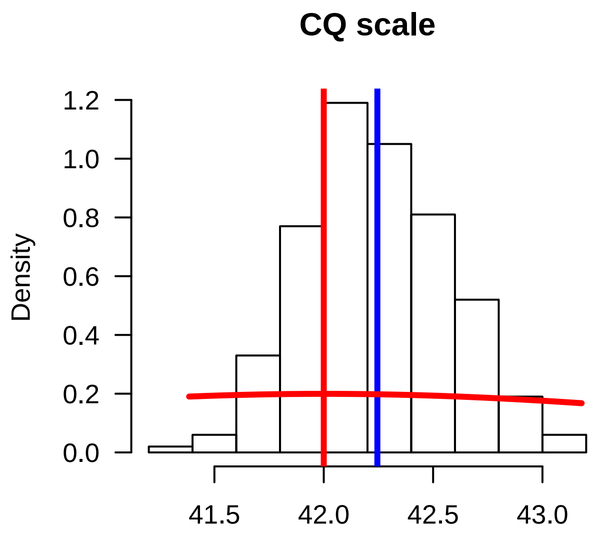

Logit c1 CQ
Density
0
1
2
3
4
−0.5
−0.3
−0.1
0.0
0.1

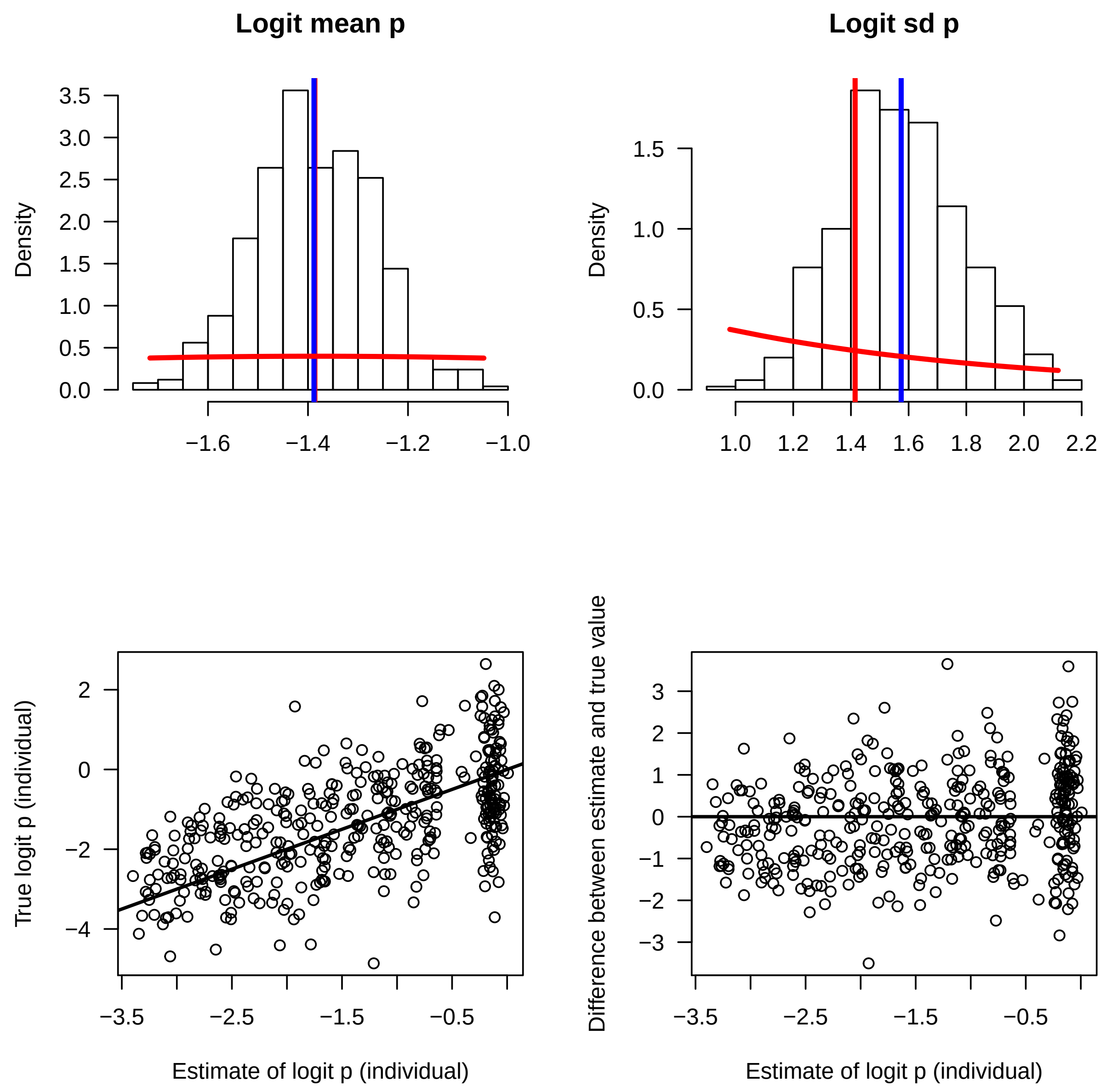

Recrudescence rate
Density
0
10
20
30
40
50
60
70
0.07
0.08
0.09
0.10
0.11


AS shape
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
7
8
9
10
11


AS scale
Density
0.0
0.5
1.0
1.5
2.0
20.2
20.6
21.0
21.4

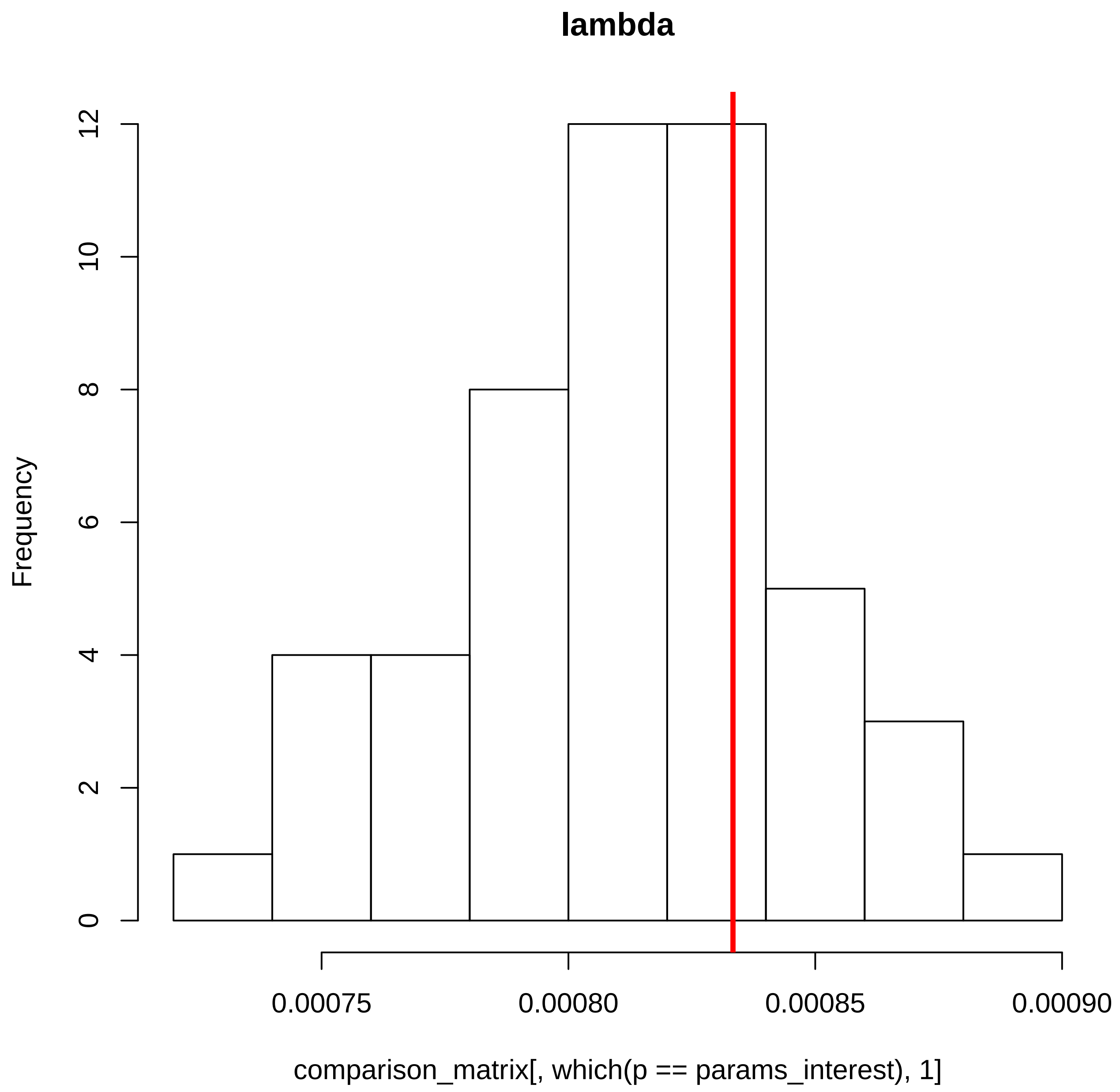

CQ shape
Density
0.0
0.1
0.2
0.3
0.4
0.5
9
10
11
12

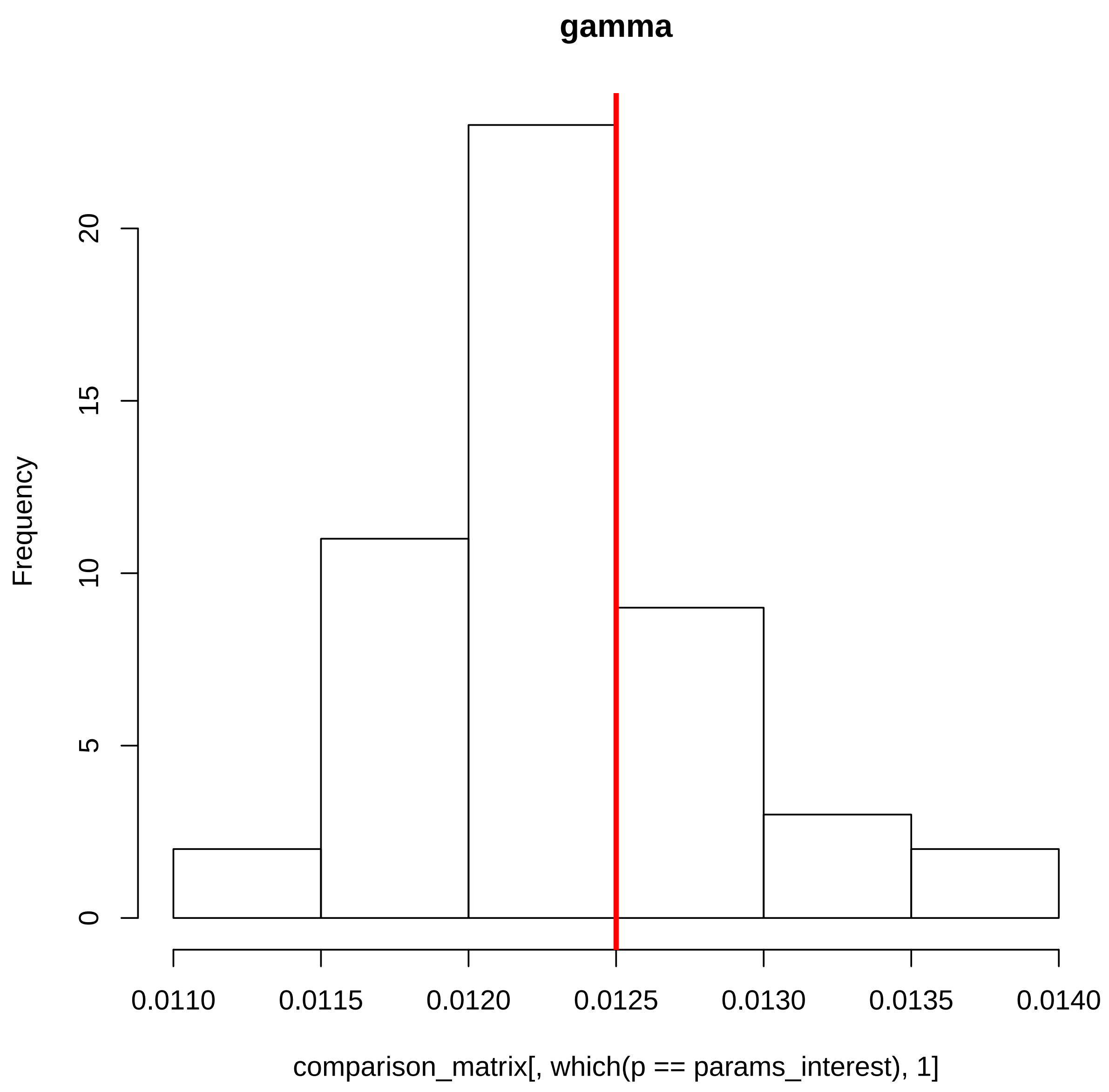

CQ scale
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
41.5
42.0
42.5
43.0

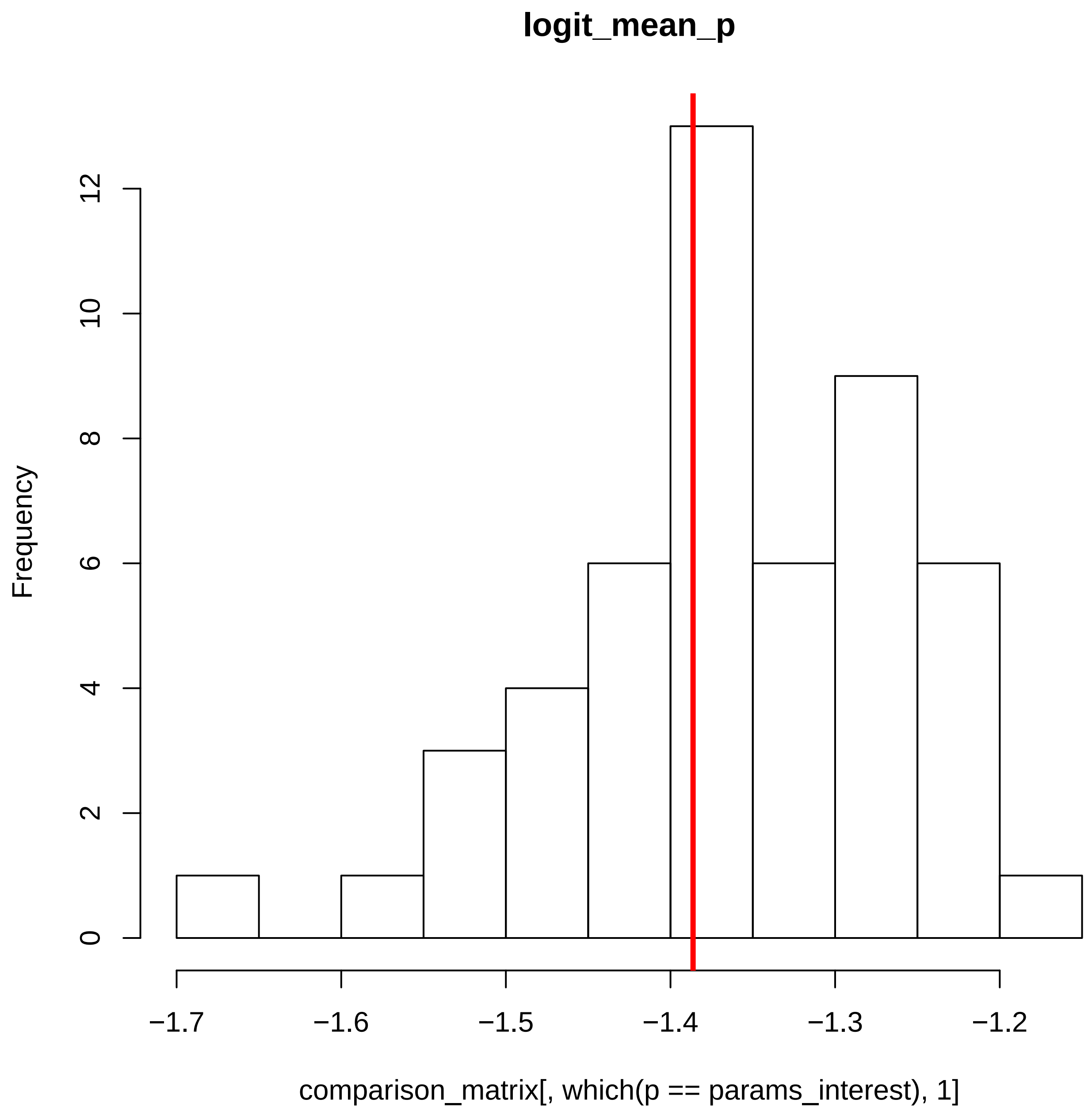


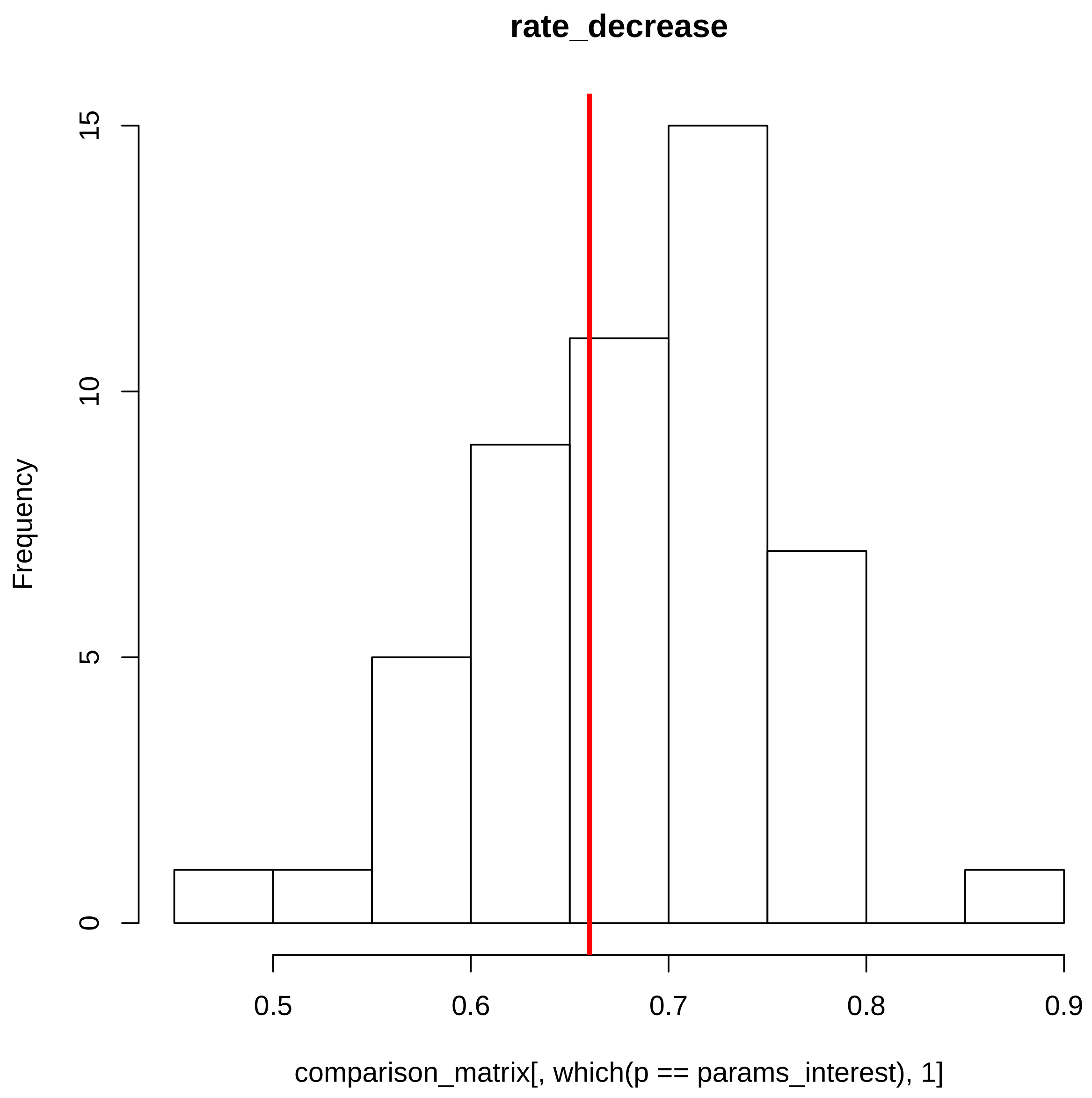
Multiple random simulations to assess systematic bias

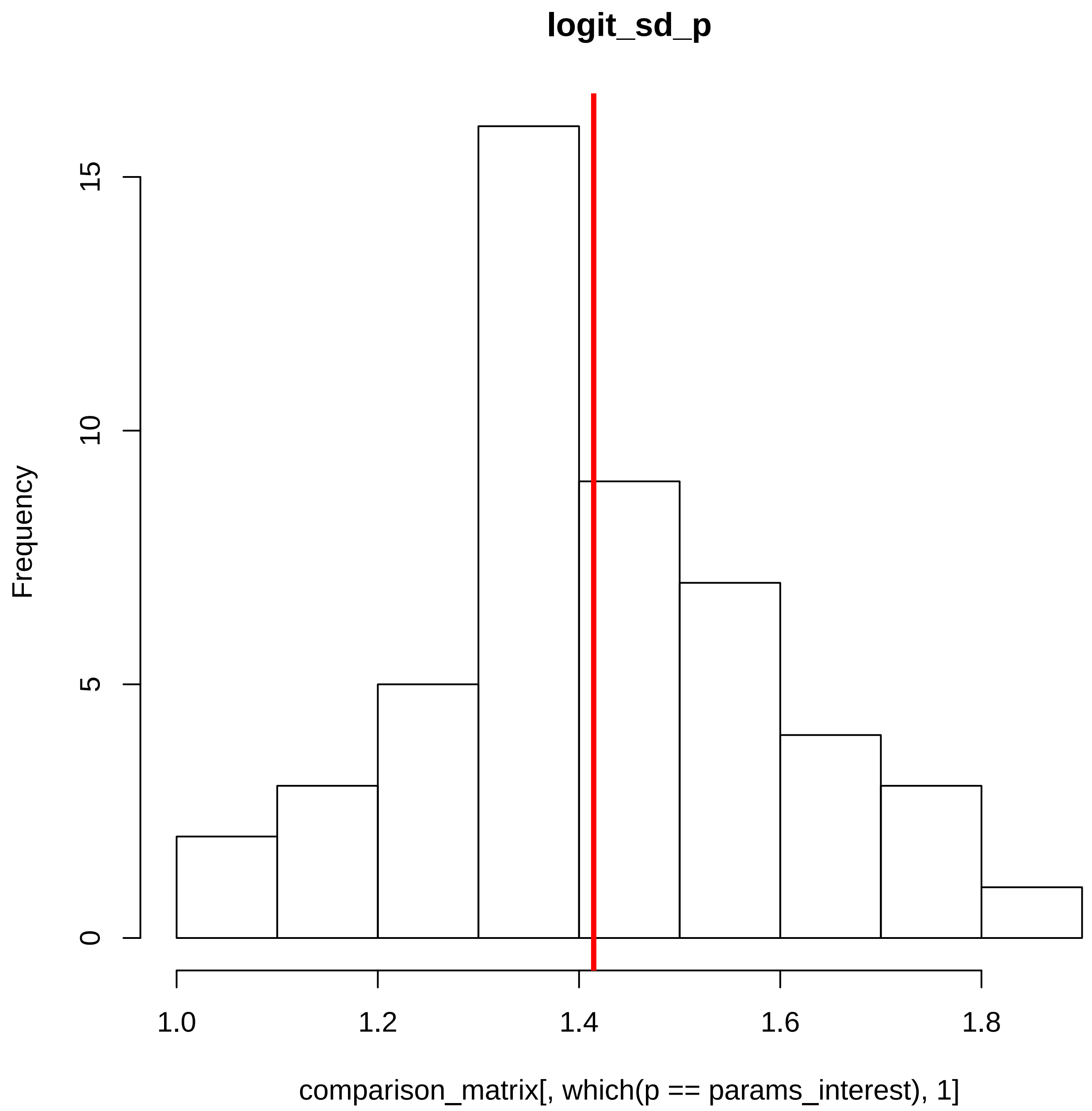
```r
for(p in params_interest){
  hist(comparison_matrix[,which(p==params_interest),1], main=p)
  abline(v=unique(comparison_matrix[,which(p==params_interest),2]),
         lwd=3,col='red')
}
```

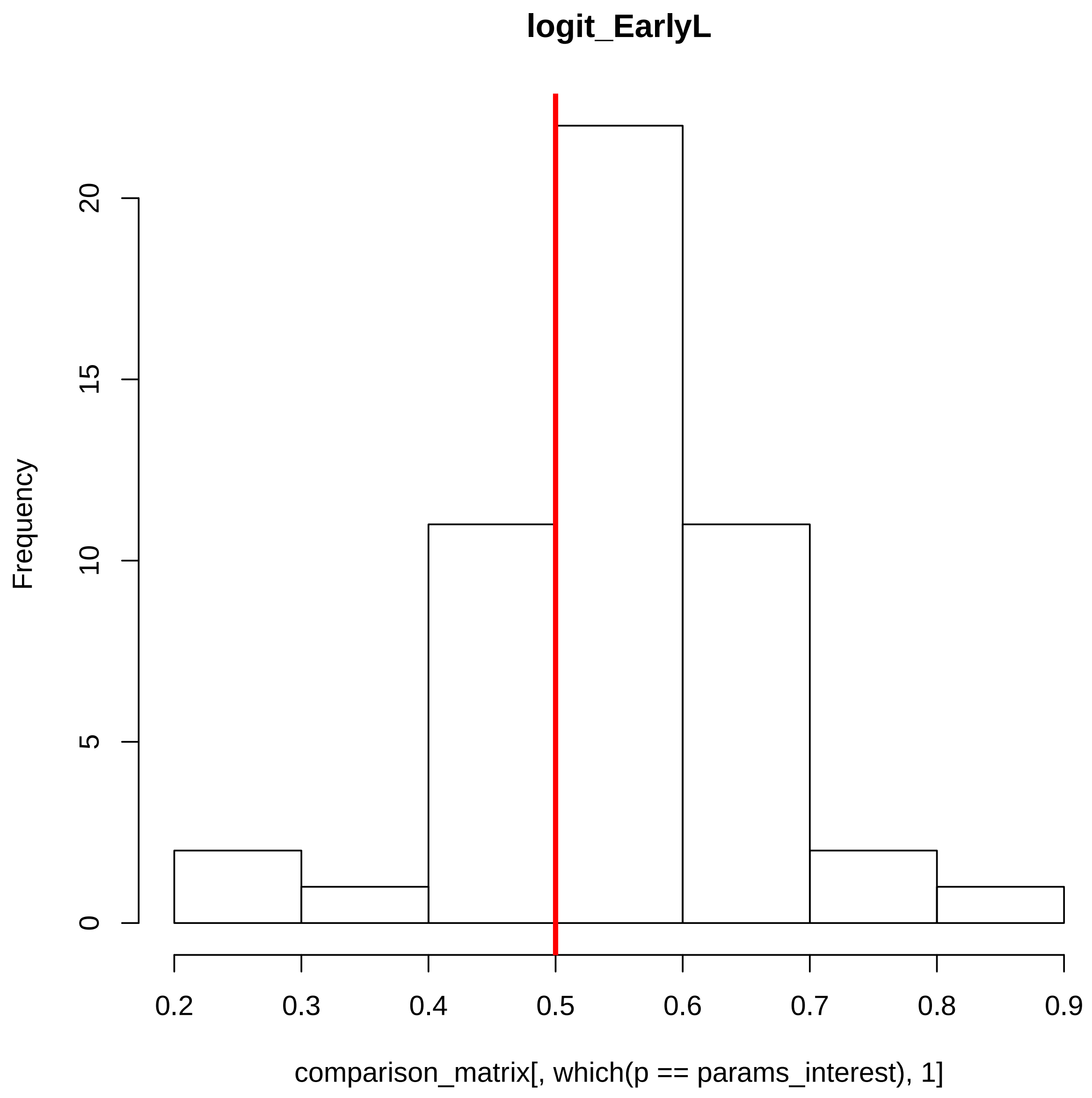

lambda
Frequency
0
2
4
6
8
10
12
0.00075
0.00080
0.00085
0.00090
comparison_matrix[, which(p == params_interest), 1]


gamma
Frequency
0
5
10
15
20
0.0110
0.0115
0.0120
0.0125
0.0130
0.0135
0.0140
comparison_matrix[, which(p == params_interest), 1]


logit_mean_p
Frequency
0
2
4
6
8
10
12
−1.7
−1.6
−1.5
−1.4
−1.3
−1.2
comparison_matrix[, which(p == params_interest), 1]


rate_decrease
Frequency
0
5
10
15
0.5
0.6
0.7
0.8
0.9
comparison_matrix[, which(p == params_interest), 1]


logit_sd_p
Frequency
0
5
10
15
1.0
1.2
1.4
1.6
1.8
comparison_matrix[, which(p == params_interest), 1]



Data simulation for model 2

Model 2 does not assume 100% efficacy of primaquine.

Ground truth model parameters for simulated data:

```
params_M2 = c(params_M1,
              logit_mean_p_PMQ = logit(0.95),
              logit_sd_p_PMQ = (-logit(0.8)+logit(0.99))/1.96)
```

Generate simulated data under the assumptions of model 2

```
set.seed(7656728)
out2 = simulate_dataset(N_PMQ = N_PMQ,
```

```r
                        N_CQ = N_CQ,
                        N_AS = N_AS,
                        FUP_time = FUP_time,
                        Study_Period = c(rep(1,N_PMQ/2),rep(2,N_PMQ/2 + N_AS + N_CQ)),
                        data_generation_function = generate_patient_data_Model2,
                        params = params_M2)
Simdata_Model2 = out2$Simdata
Simulation_truth2 = out2$Simulation_truth
table(Simulation_truth2$True_state, Simulation_truth2$Drug)
```

```
##
##                   AS CHQ CHQ/PMQ
##   EarlyRelapse  286 284      53
##   LateRelapse   173 145      25
##   Recrudescence 401 407      80
##   Reinfection   167 168     946
```

Prior specification for stan fit:

```r
Prior_params_M2 = c(Prior_params_M1,
                    Hyper_logit_mean_pPMQ_mean = logit(0.95),
                    Hyper_logit_mean_pPMQ_sd = (logit(0.95)-logit(0.7))/1.96,
                    Hyper_logit_sd_pPMQ_lambda = 1)
```

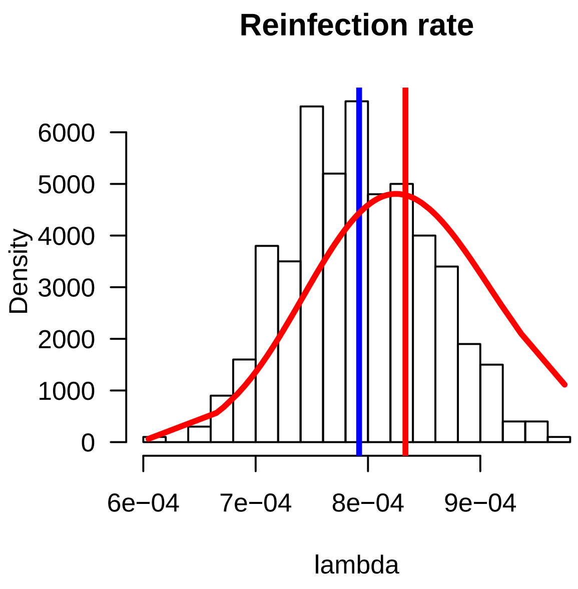
Load or compile the stan model 2

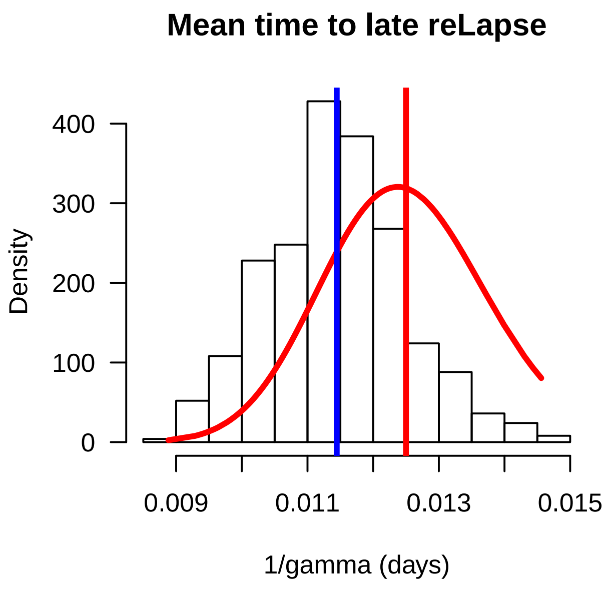
```r
if(RECOMPILE_MODELS){
  writeLines('Compiling model 2....')
  source('../Timing_Model/StanModel2.R')
  save(Timing_Model2, file = '../RData/TimingModel/Timing_Model2.RData')
} else {
  load('../RData/TimingModel/Timing_Model2.RData')
}
```

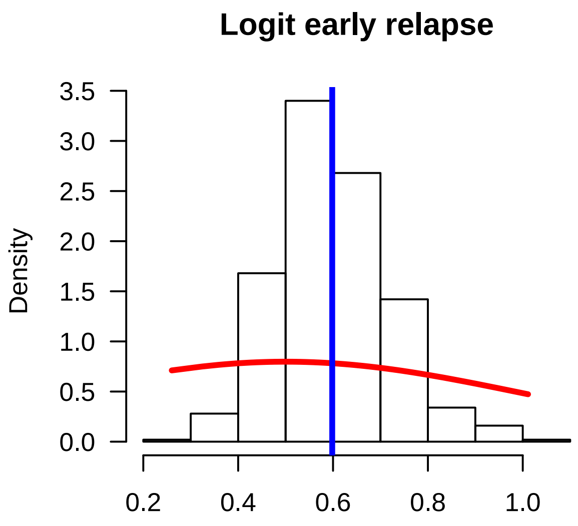
Fit stan model 2 to simulated data from model 2

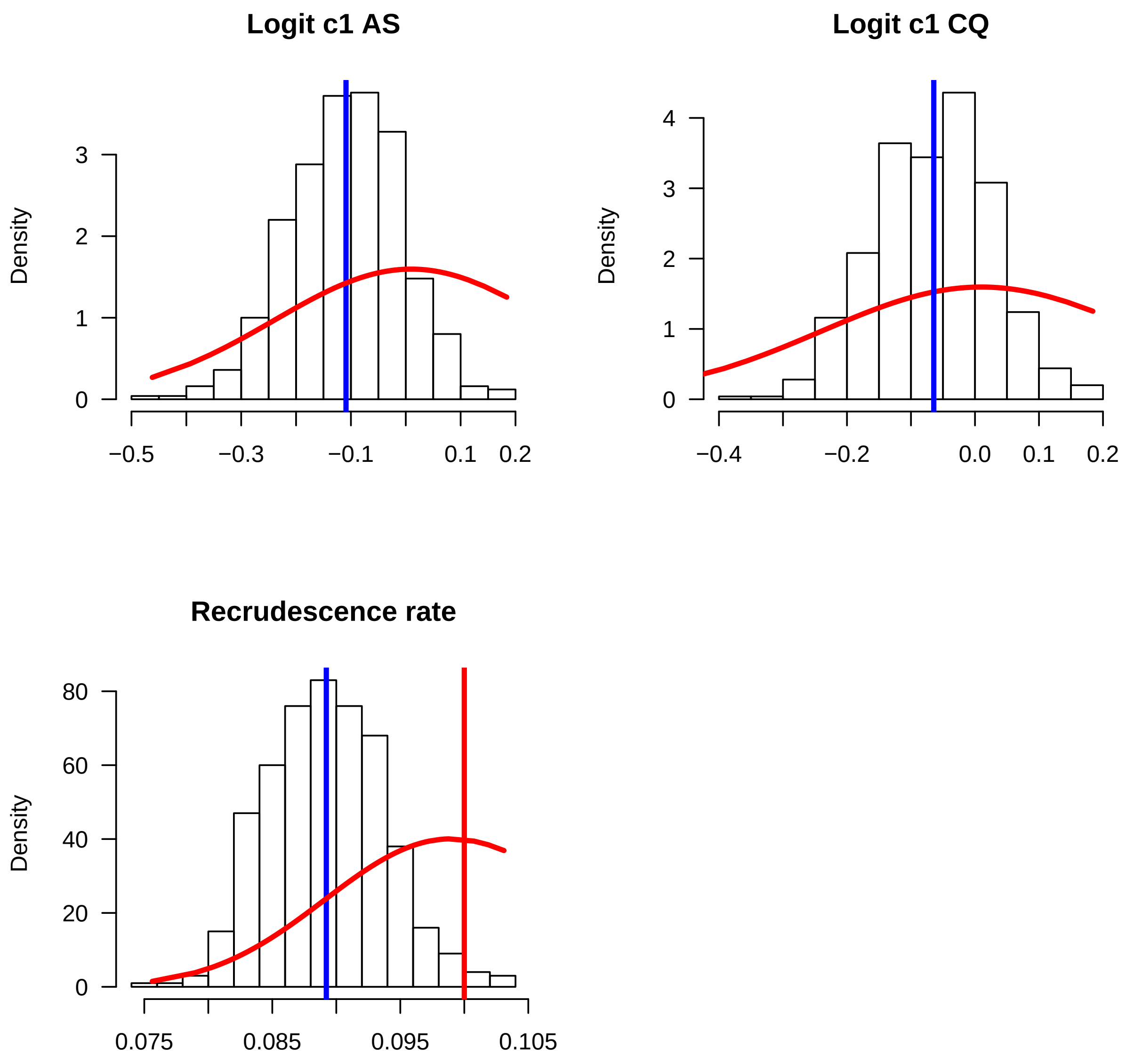
Plot output, comparing prior, ground truth and model estimate:

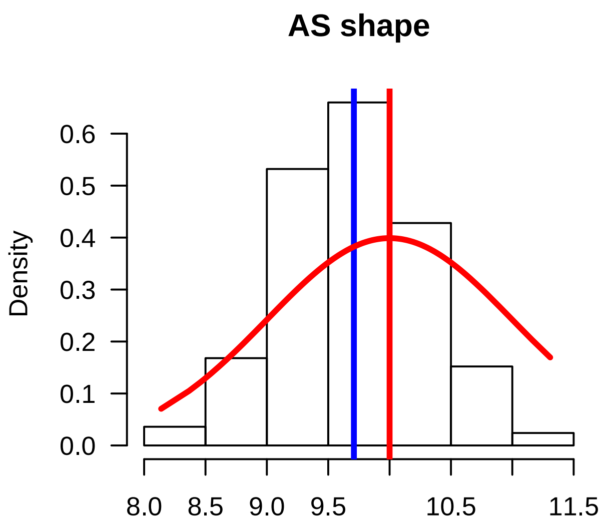
```r
plot_output_model2(thetas_mod2,Simulation_truth2,Simdata_Model2,Prior_params_M2)
```

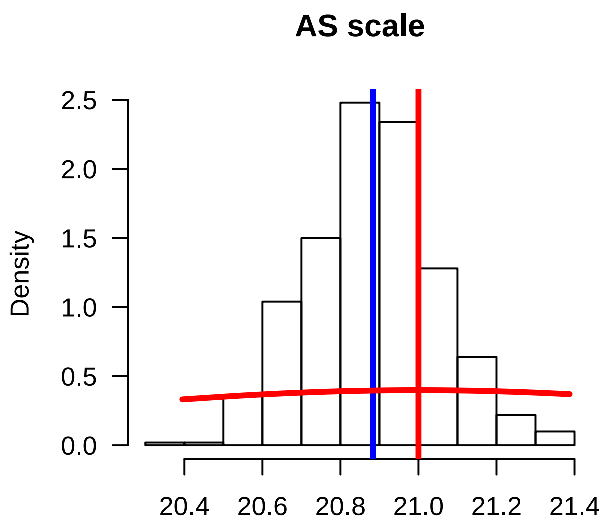

Reinfection rate
Density
6000
5000
4000
3000
2000
1000
0
6e−04
7e−04
8e−04
9e−04
lambda

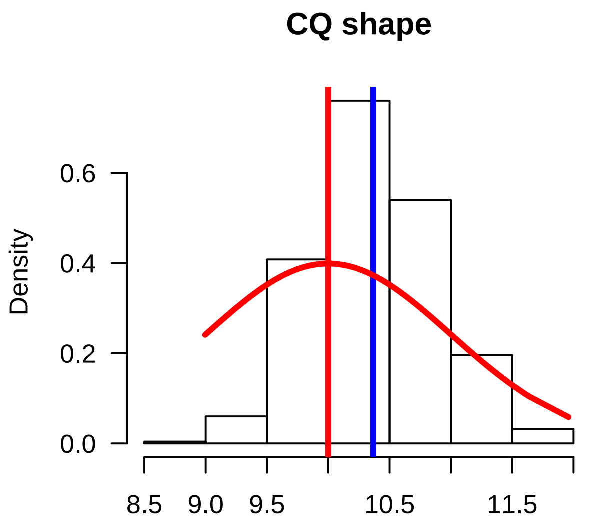

Mean time to late reLapse
Density
400
300
200
100
0
0.009
0.011
0.013
0.015
1/gamma (days)

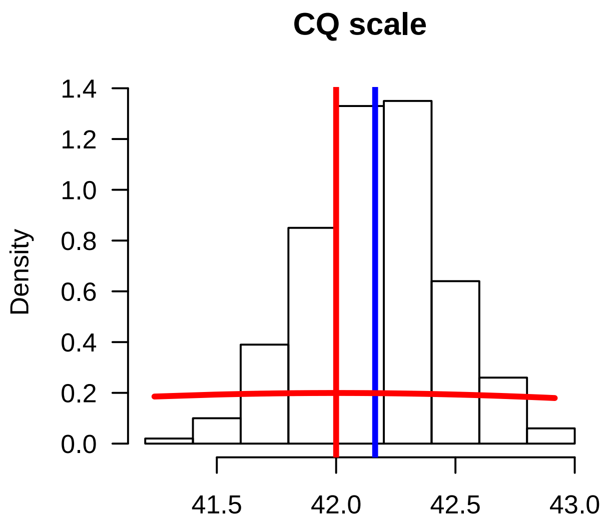

Logit early relapse
Density
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0

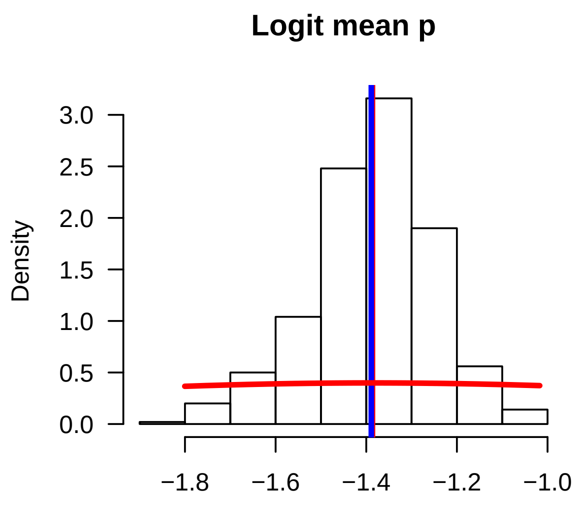
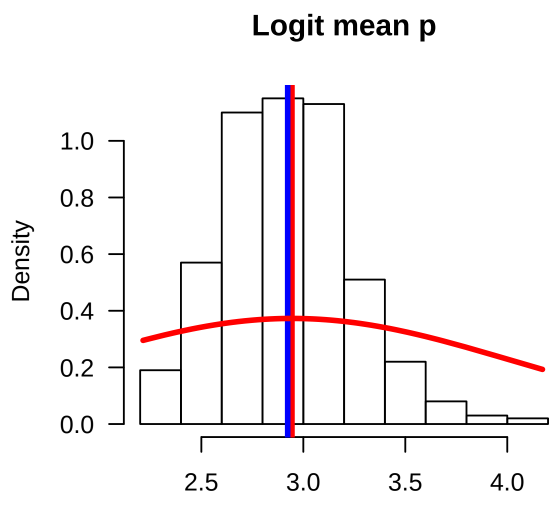
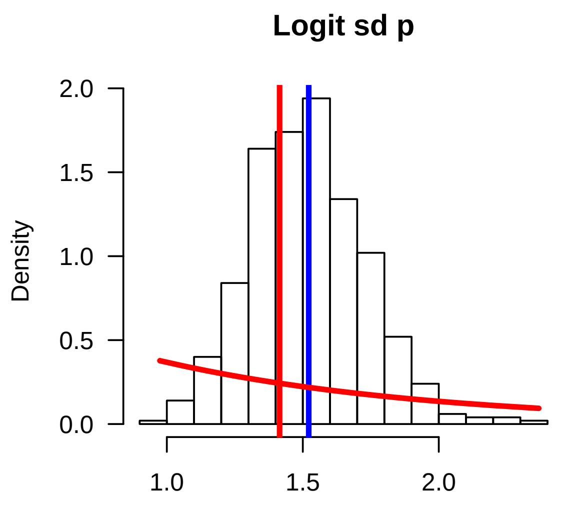
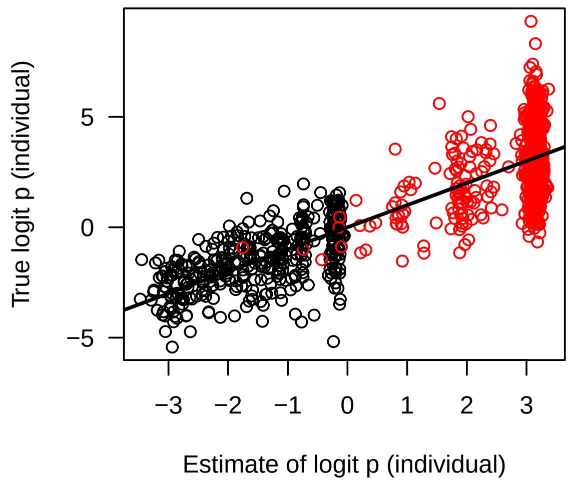

Logit c1 AS
Density
−0.5
−0.3
−0.1
0.1
0.2
0
1
2
3
Logit c1 CQ
Density
−0.4
−0.2
0.0
0.1
0.2
0
1
2
3
4
Recrudescence rate
Density
0.075
0.085
0.095
0.105
0
20
40
60
80

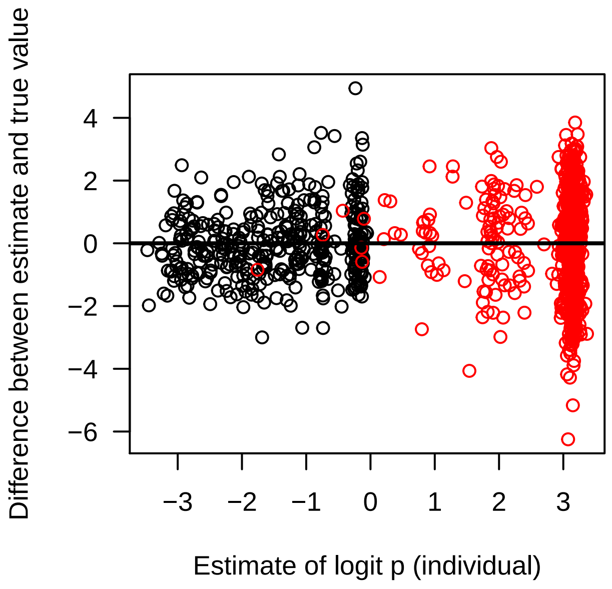

AS shape
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
8.0
8.5
9.0
9.5
10.5
11.5


AS scale
Density
0.0
0.5
1.0
1.5
2.0
2.5
20.4
20.6
20.8
21.0
21.2
21.4


CQ shape
Density
0.0
0.2
0.4
0.6
8.5
9.0
9.5
10.5
11.5


CQ scale
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
41.5
42.0
42.5
43.0


Logit mean p
Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1.8
−1.6
−1.4
−1.2
−1.0

Logit mean p
Density
1.0
0.8
0.6
0.4
0.2
0.0
2.5
3.0
3.5
4.0

Logit sd p
Density
2.0
1.5
1.0
0.5
0.0
1.0
1.5
2.0

True logit p (individual)
5
0
−5
−3
−2
−1
0
1
2
3
Estimate of logit p (individual)



Compare summary statistics from simulated data and estimated:

```
# the true failure rate in the primaquine arm
sum(Simulation_truth2$True_state[Simulation_truth2$Drug=="CHQ/PMQ"] != 'Reinfection')/sum(Simulation_tru
```

```
## [1] 0.1431159
```

```
classification_mod2 = apply(thetas_mod2$prob_labels,c(2,3),quantile, probs=0.975)
1-sum(classification_mod2[Simdata_Model2$Drug==2,1])/sum(Simdata_Model2$Drug==2)
```

```
## [1] 0.07240821
```

```
classification_mod2 = apply(thetas_mod2$prob_labels,c(2,3),median)
1-sum(classification_mod2[Simdata_Model2$Drug==2,1])/sum(Simdata_Model2$Drug==2)
```

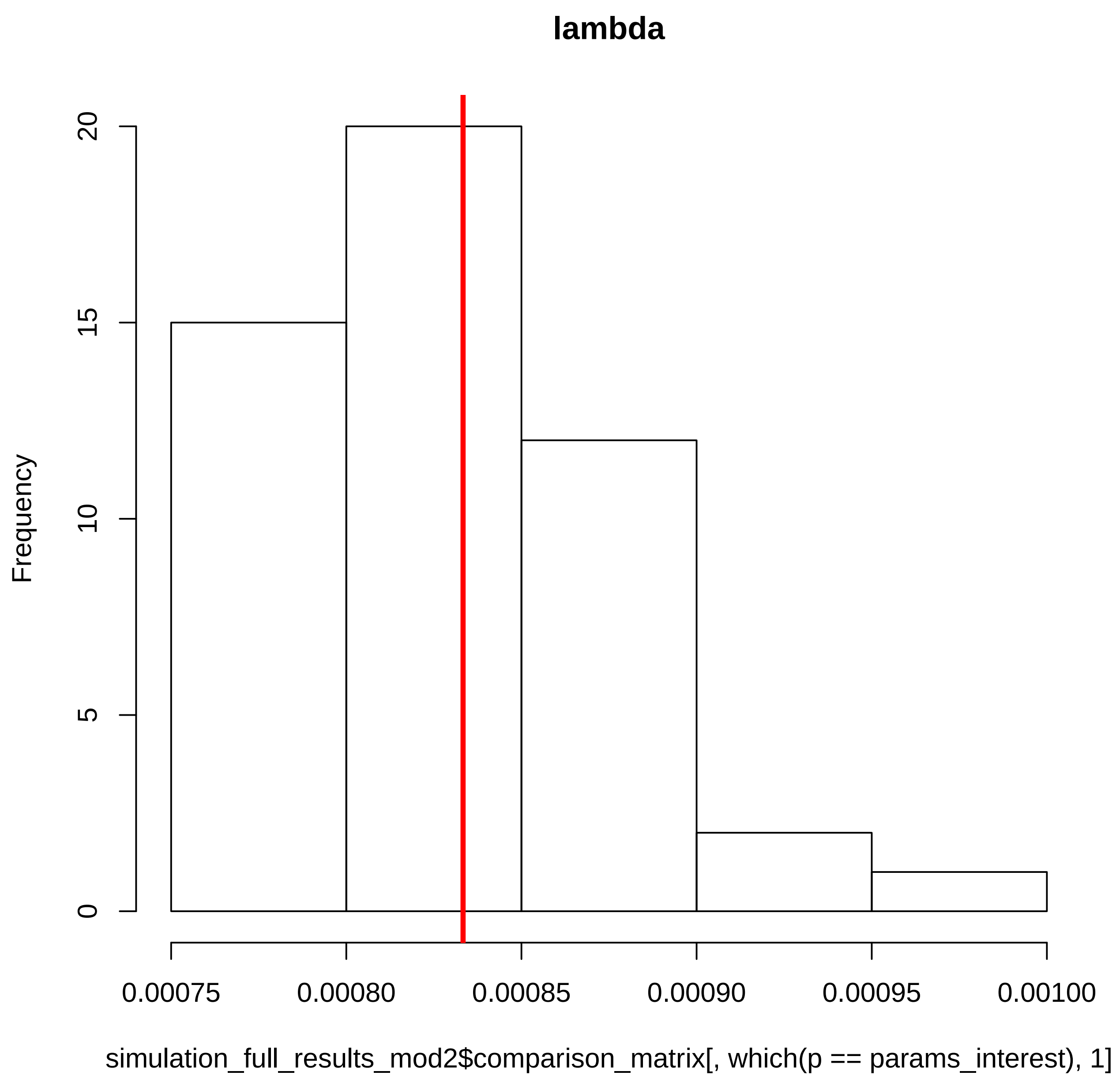
```
## [1] 0.1491698
```

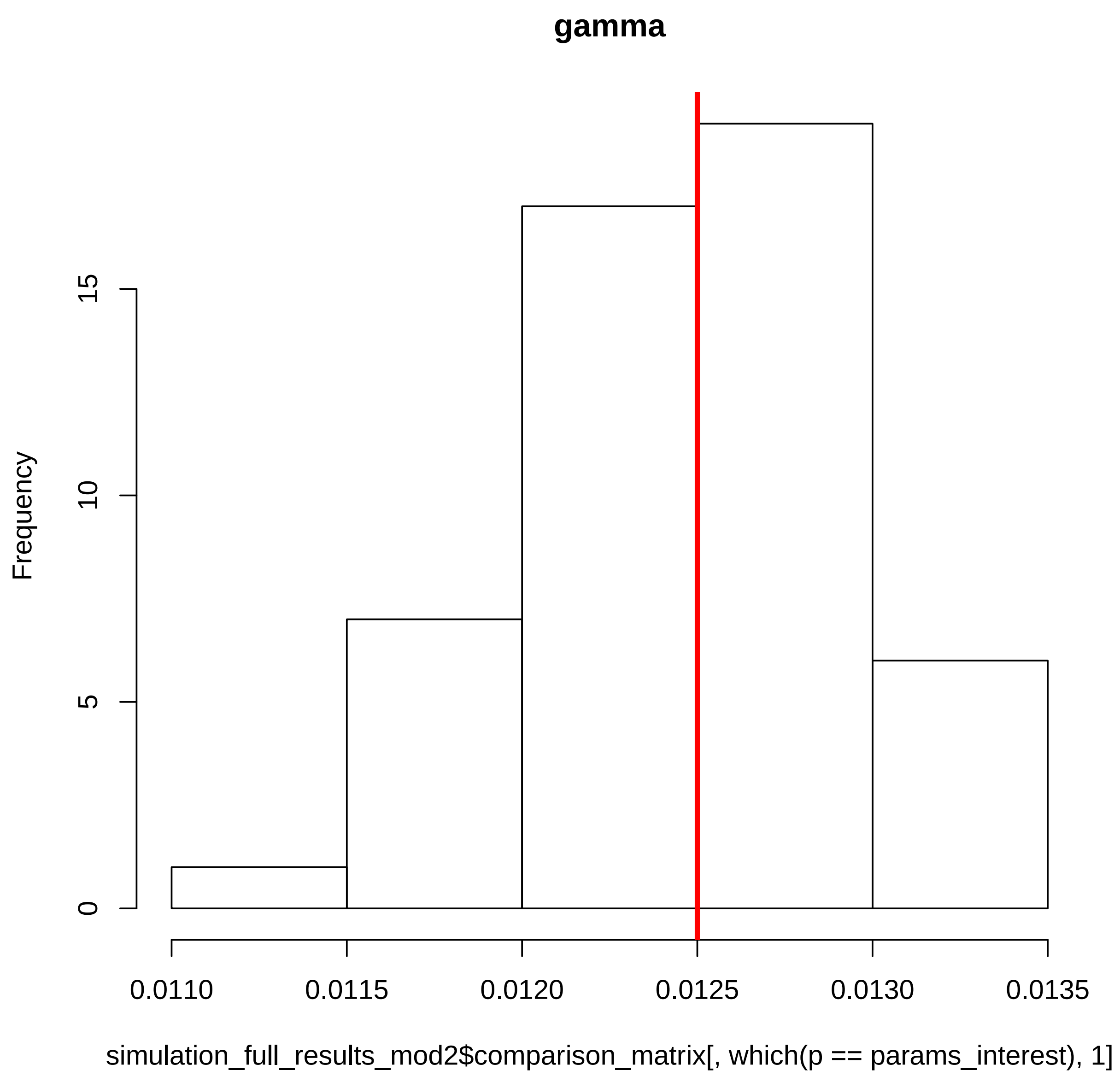
```
classification_mod2 = apply(thetas_mod2$prob_labels,c(2,3),quantile, probs=0.025)
1-sum(classification_mod2[Simdata_Model2$Drug==2,1])/sum(Simdata_Model2$Drug==2)
```

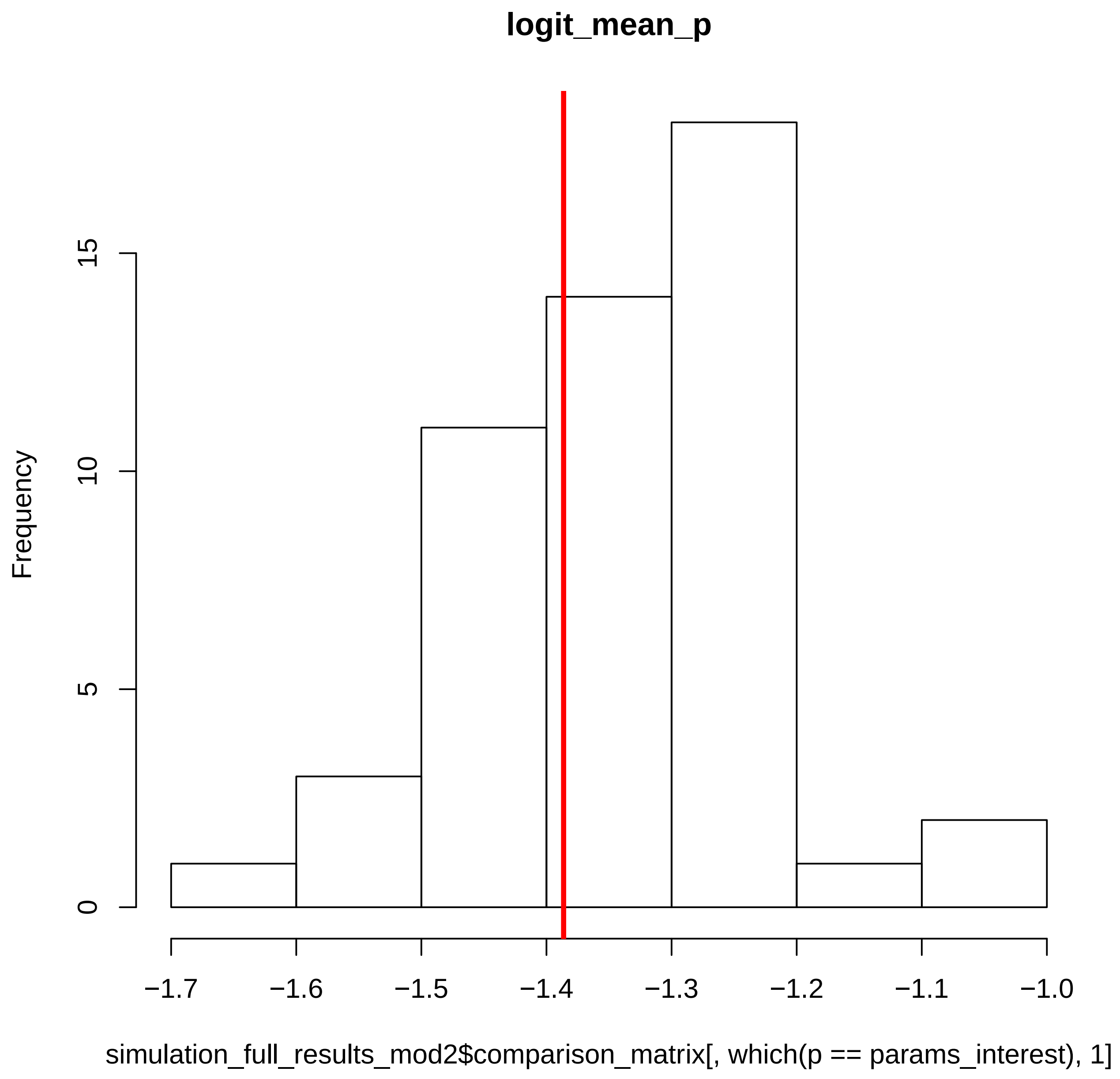
```
## [1] 0.2166946
```

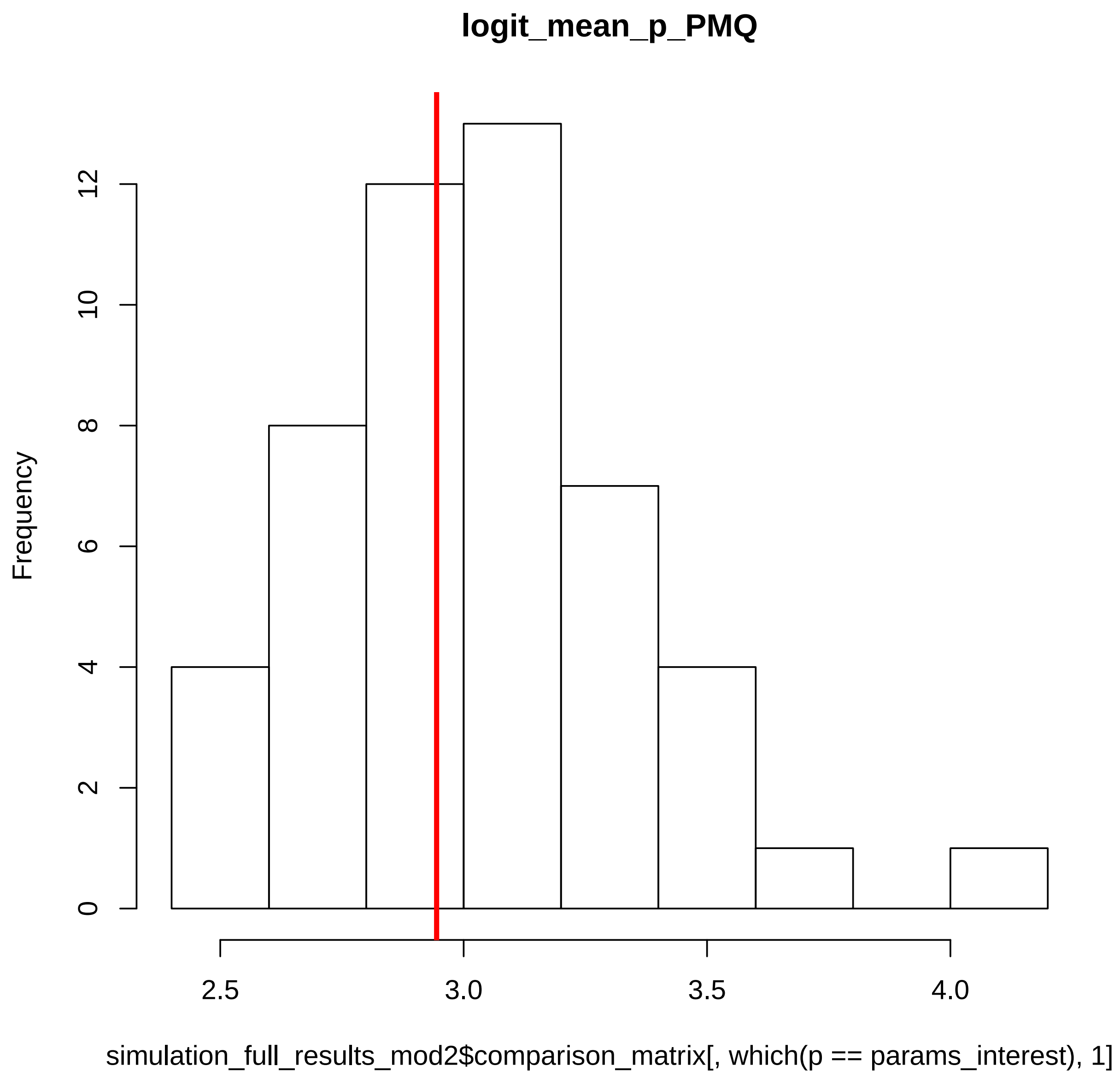
Multiple random simulations to assess systematic bias

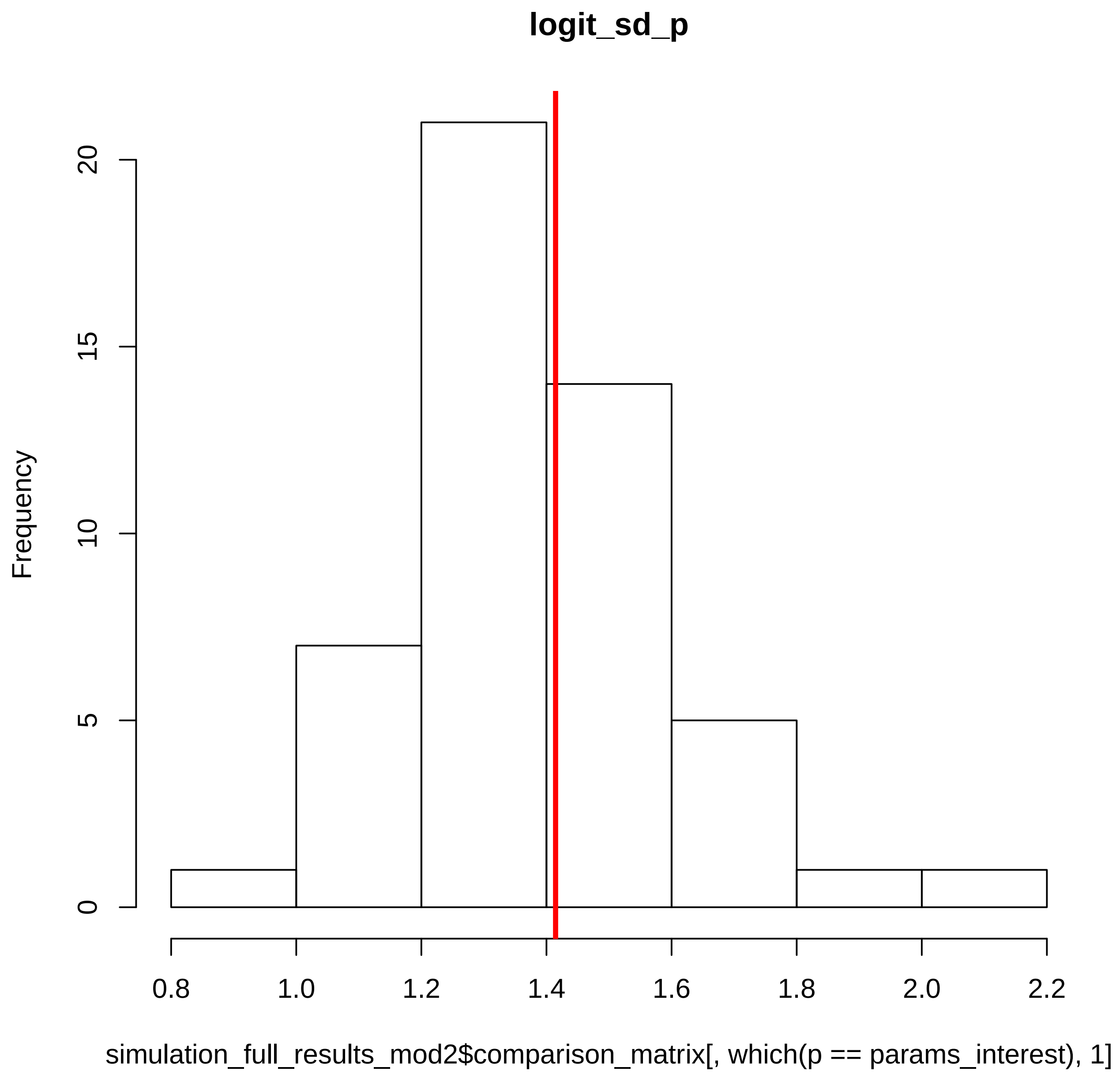
```
for(p in params_interest){
  hist(simulation_full_results_mod2$comparison_matrix[,which(p==params_interest),1], main=p)
  abline(v=unique(simulation_full_results_mod2$comparison_matrix[,which(p==params_interest),2]),col='re
}
```

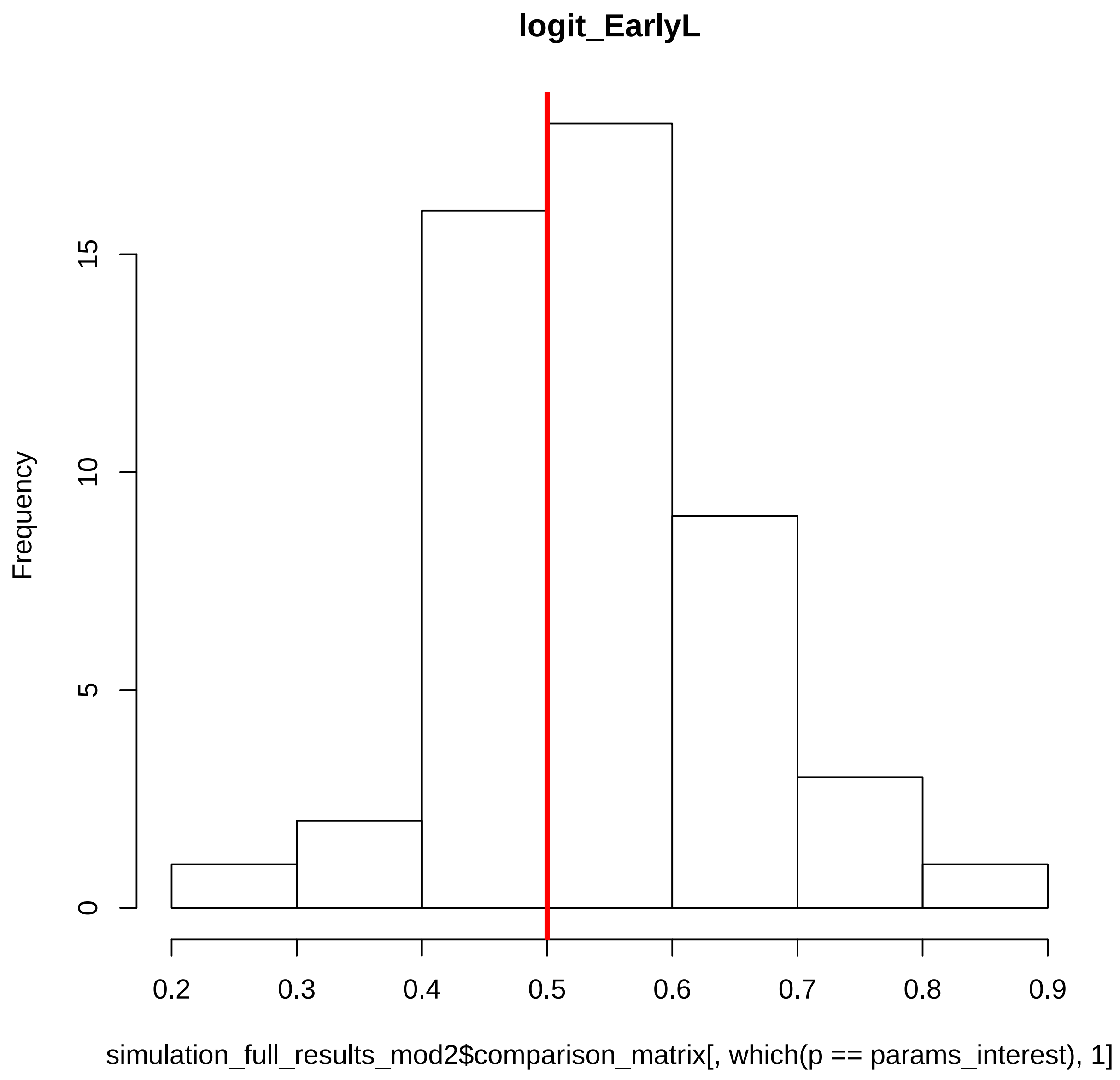

lambda
Frequency
0
5
10
15
20
0.00075
0.00080
0.00085
0.00090
0.00095
0.00100
simulation_full_results_mod2$comparison_matrix[, which(p == params_interest), 1]


gamma
Frequency
0
5
10
15
0.0110
0.0115
0.0120
0.0125
0.0130
0.0135
simulation_full_results_mod2$comparison_matrix[, which(p == params_interest), 1]

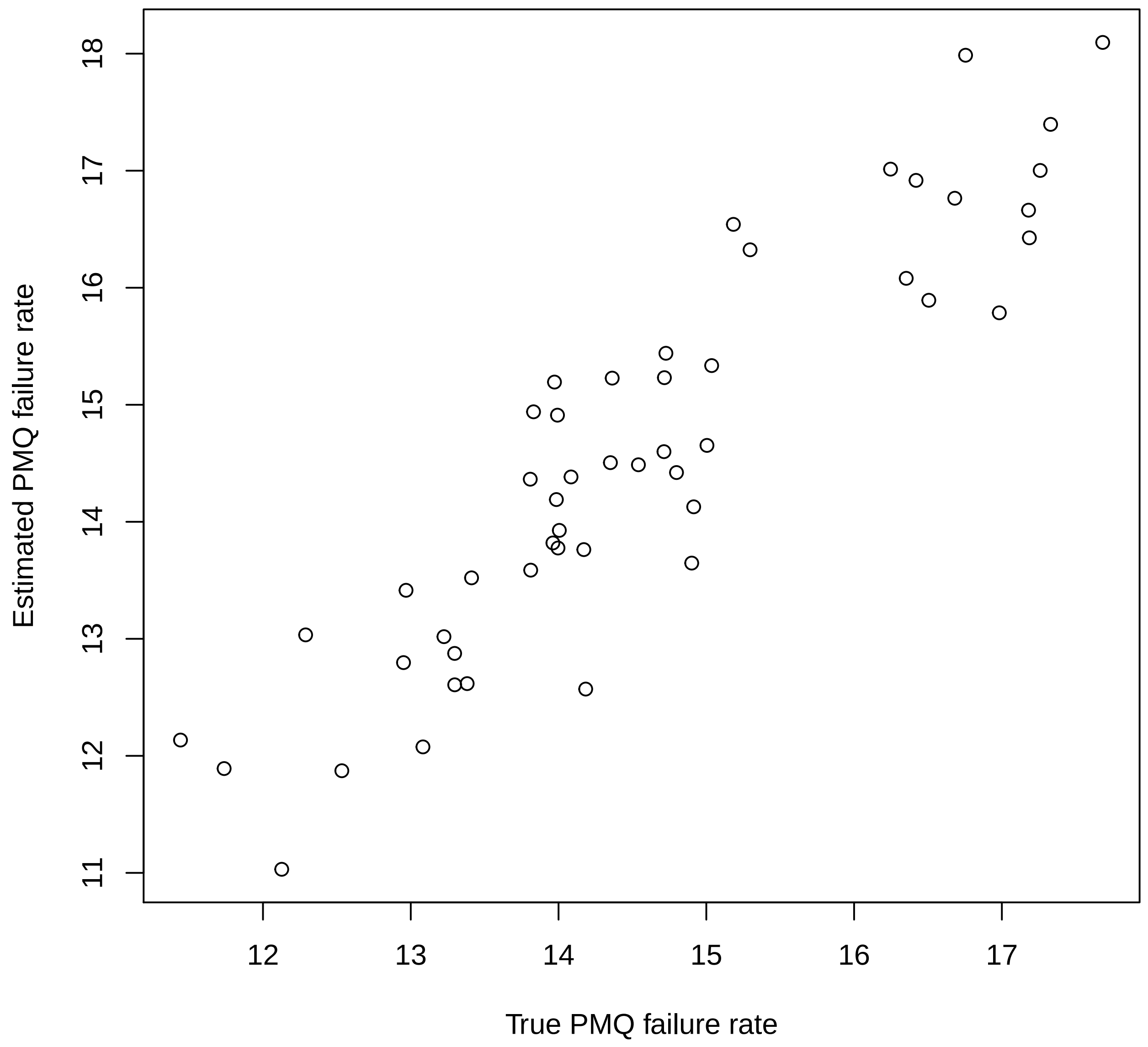

logit_mean_p
Frequency
0
5
10
15
−1.7
−1.6
−1.5
−1.4
−1.3
−1.2
−1.1
−1.0
simulation_full_results_mod2$comparison_matrix[, which(p == params_interest), 1]


logit_mean_p_PMQ
Frequency
0
2
4
6
8
10
12
2.5
3.0
3.5
4.0
simulation_full_results_mod2$comparison_matrix[, which(p == params_interest), 1]


logit_sd_p
Frequency
0
5
10
15
20
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
simulation_full_results_mod2$comparison_matrix[, which(p == params_interest), 1]



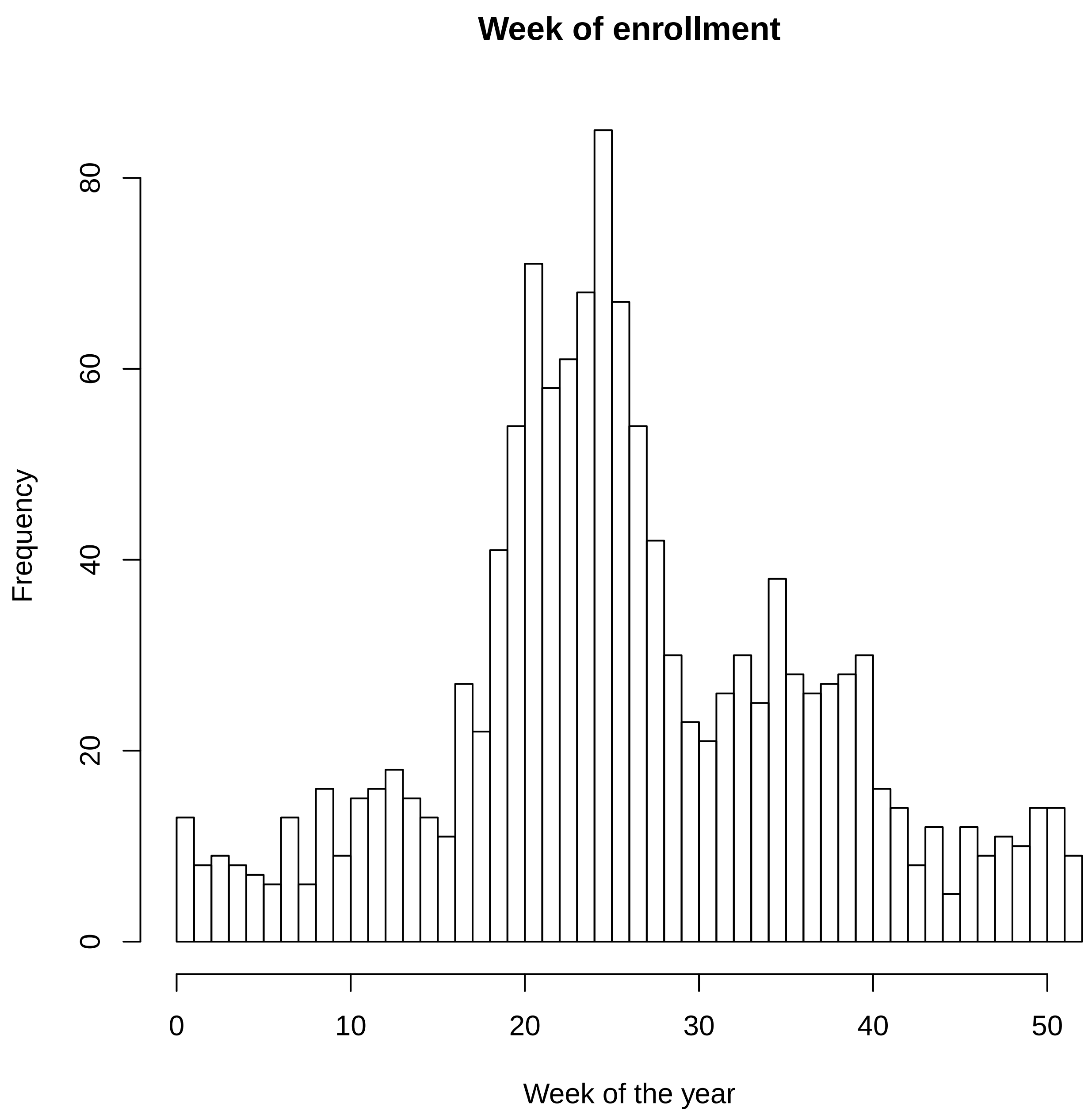
```
plot(100*simulation_full_results_mod2$failure_matrix[,1],
     100*simulation_full_results_mod2$failure_matrix[,2],
     xlab='True PMQ failure rate', ylab = 'Estimated PMQ failure rate')
lines(c(0,1),c(0,1))
```



Simulating seasonal variation

Estimate the empirical seasonal reinfecton distribution from enrollment episodes. This assumes that the majority of enrollment episodes are reinfections which should be approximately true.

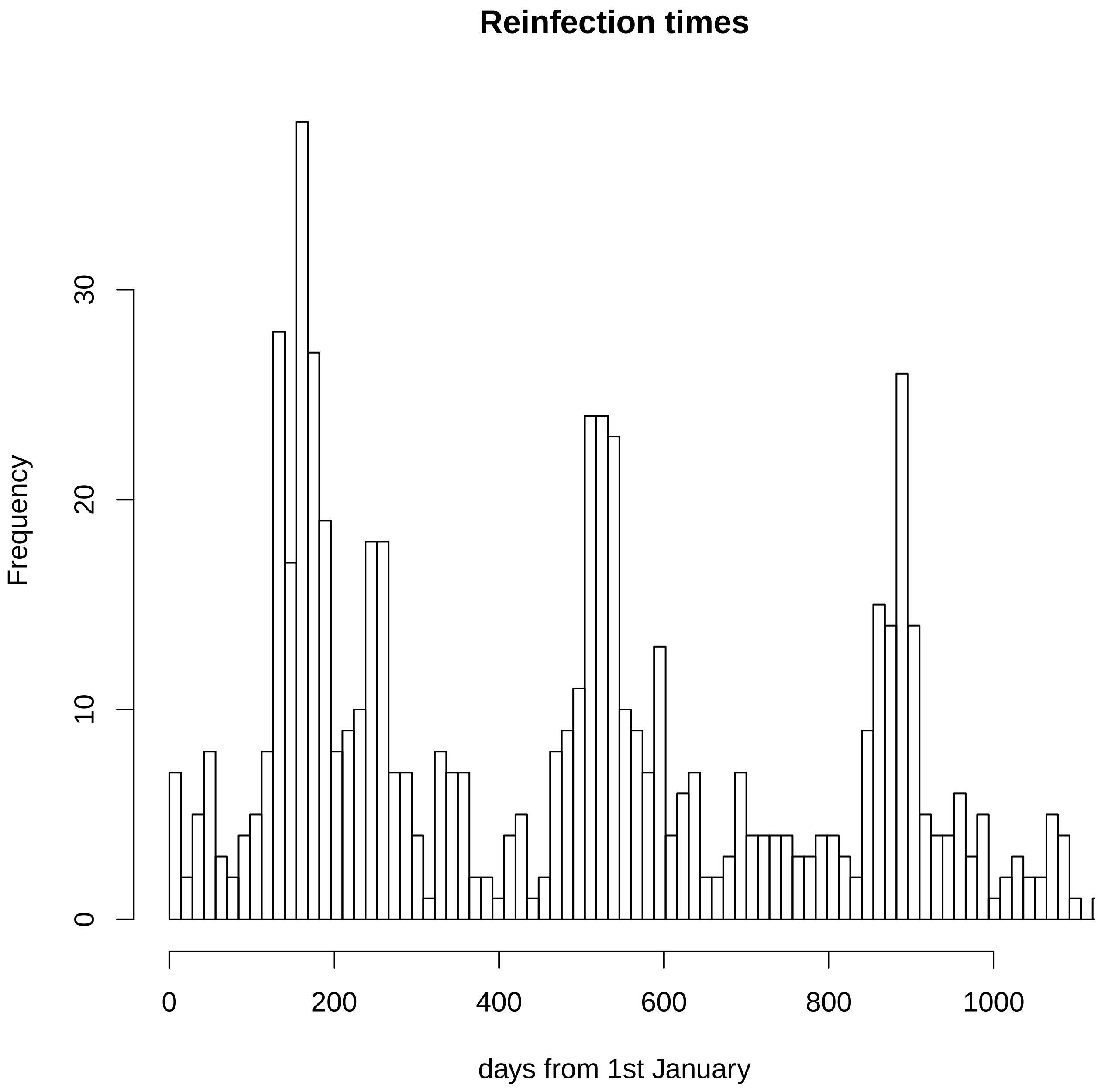
```r
load('../RData/TimingModel/Combined_Time_Event.RData')
Combined_Time_Data = filter(Combined_Time_Data, Censored == -1)
ind = Combined_Time_Data$WeekTime>52
Combined_Time_Data$WeekTime[ind] = Combined_Time_Data$WeekTime[ind]-52
hist(Combined_Time_Data$WeekTime, breaks = 0:52,
     main = 'Week of enrollment',xlab='Week of the year')
```



```
# the vector we can use for empirical distribution sampling
seasonal_sampling_vector = Combined_Time_Data$WeekTime
```

Generate patient data under the assumptions of model 1 with the additional assumption of seasonality for reinfection.

```
# test to show this does the right thing
set.seed(7576)
ys = sapply(1:1000, FUN = function(x, seasonal_sampling_vector){
  generate_reinfection_time_seasonal(params = params_M1,
                                     week_time = 0,
                                     seasonal_sampling_vector = seasonal_sampling_vector,
                                     Study_Period = 1)
}, seasonal_sampling_vector)
hist(ys, breaks = seq(0,20000,by=14), xlim = c(0,3*360),
     main='Reinfection times',
     xlab='days from 1st January')
```



Generate data with these parameters:

```
set.seed(87678)
tic()
out3 = simulate_dataset(N_PMQ = N_PMQ, N_CQ = N_CQ,
                        N_AS = N_AS,FUP_time = FUP_time,
                        data_generation_function = generate_patient_data_Model2_Seasonal,
                        params = params_M2,
                        seasonal_sampling_vector = seasonal_sampling_vector,
                        Study_Period = c(rep(1,N_PMQ/2),rep(2,N_PMQ/2 + N_AS + N_CQ)))
toc()
```

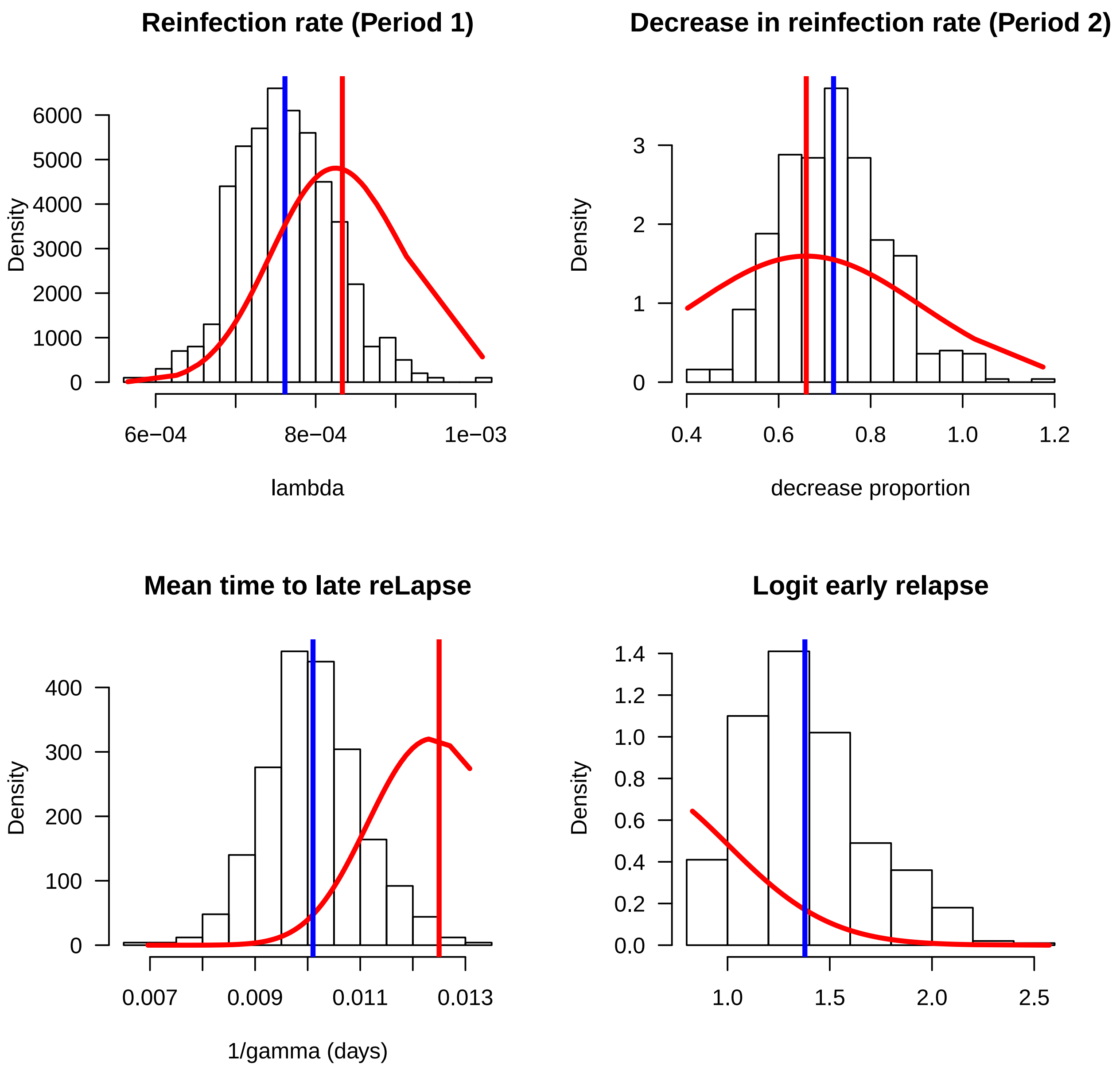
```
## 0.872 sec elapsed
```

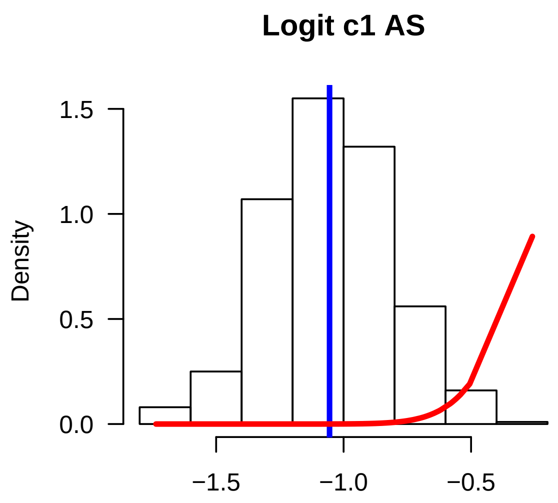
```
Simdata_Model_Seasonal = out3$Simdata
Simulation_truth_Seasonal = out3$Simulation_truth
```

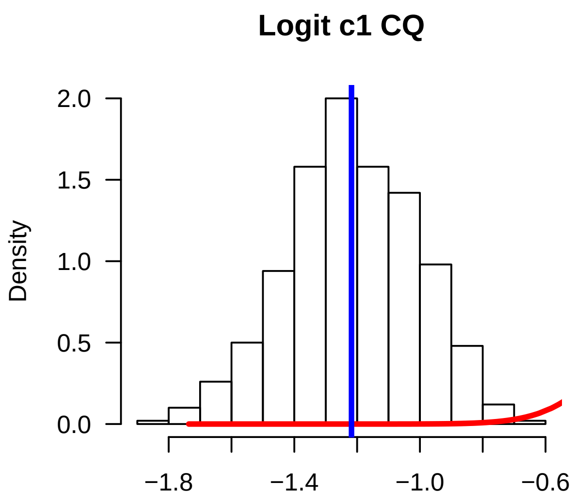
Fit stan model to simulated data where assumption of constant reinfection rate is wrong

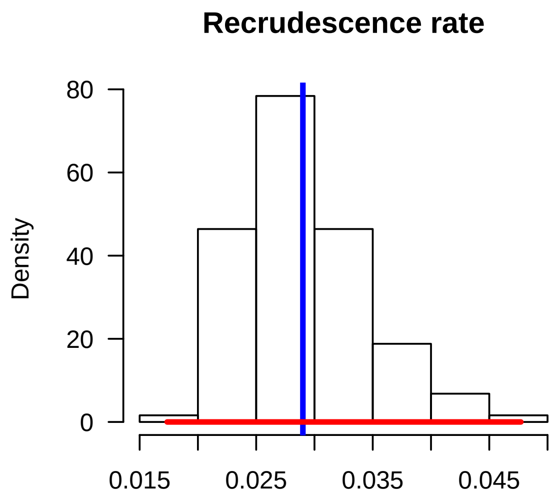
Plot output, comparing prior, ground truth and model estimate:

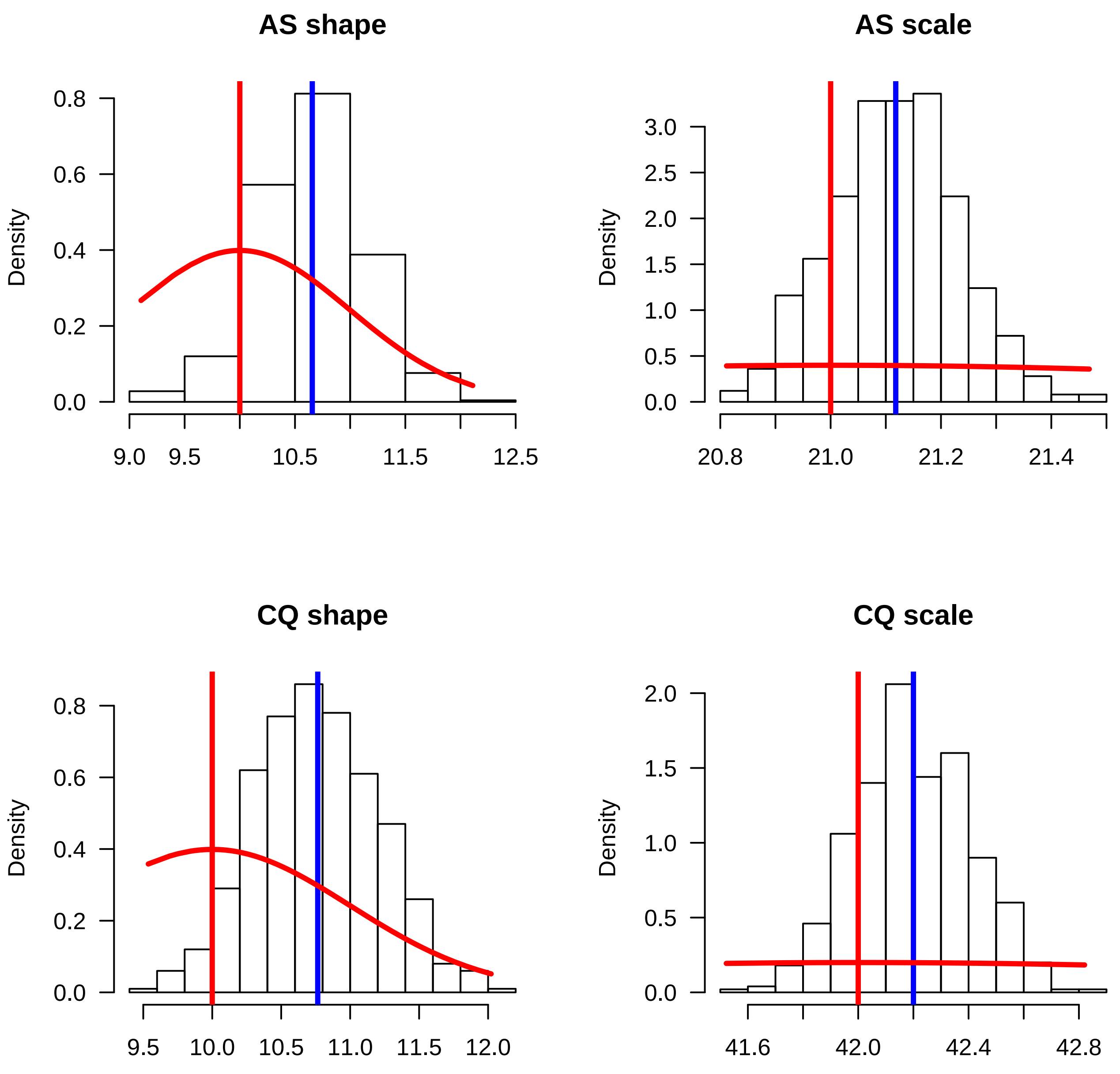
```
plot_output_model1(thetas_mod_seasonal,Simulation_truth_Seasonal,
                   Simdata_Model_Seasonal,Prior_params_M1)
```

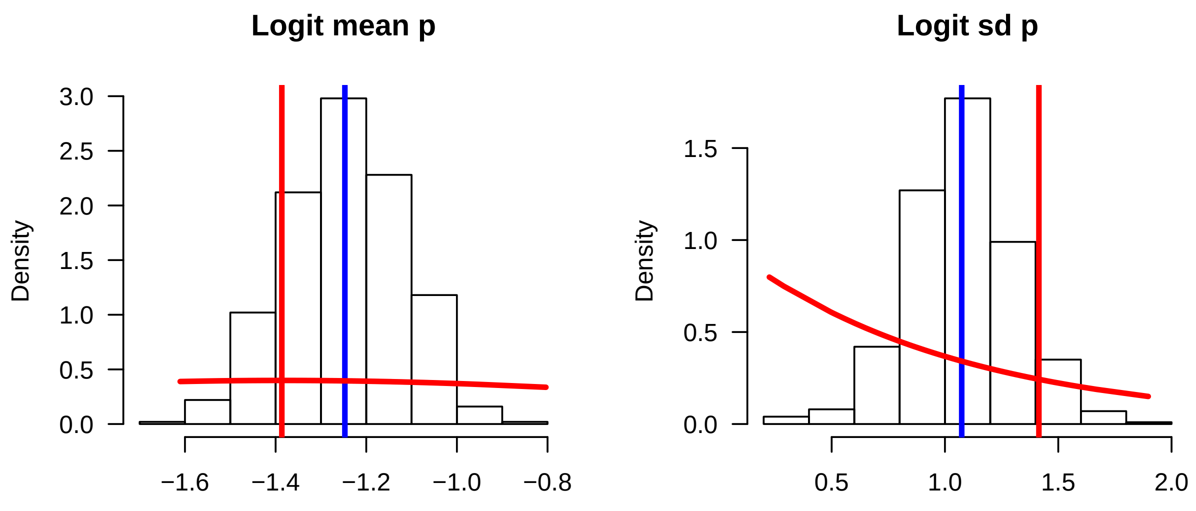


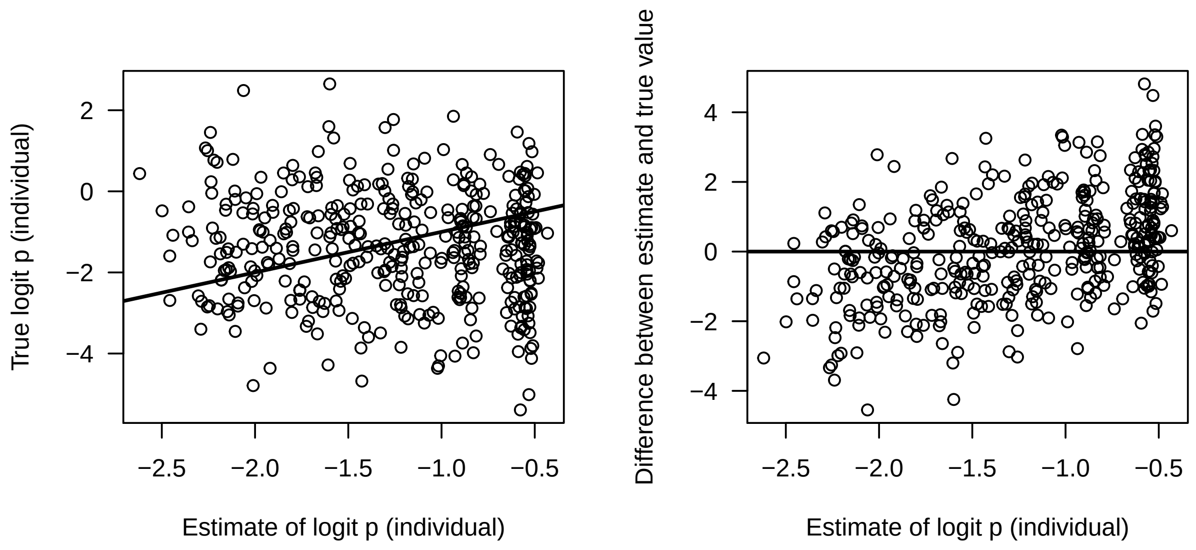

Logit c1 AS
Density
0.0
0.5
1.0
1.5
−1.5
−1.0
−0.5


Logit c1 CQ
Density
0.0
0.5
1.0
1.5
2.0
−1.8
−1.4
−1.0
−0.6


Recrudescence rate
Density
0
20
40
60
80
0.015
0.025
0.035
0.045


AS shape
Density
0.0
0.2
0.4
0.6
0.8
9.0
9.5
10.5
11.5
12.5
AS scale
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
20.8
21.0
21.2
21.4
CQ shape
Density
0.0
0.2
0.4
0.6
0.8
9.5
10.0
10.5
11.0
11.5
12.0
CQ scale
Density
0.0
0.5
1.0
1.5
2.0
41.6
42.0
42.4
42.8

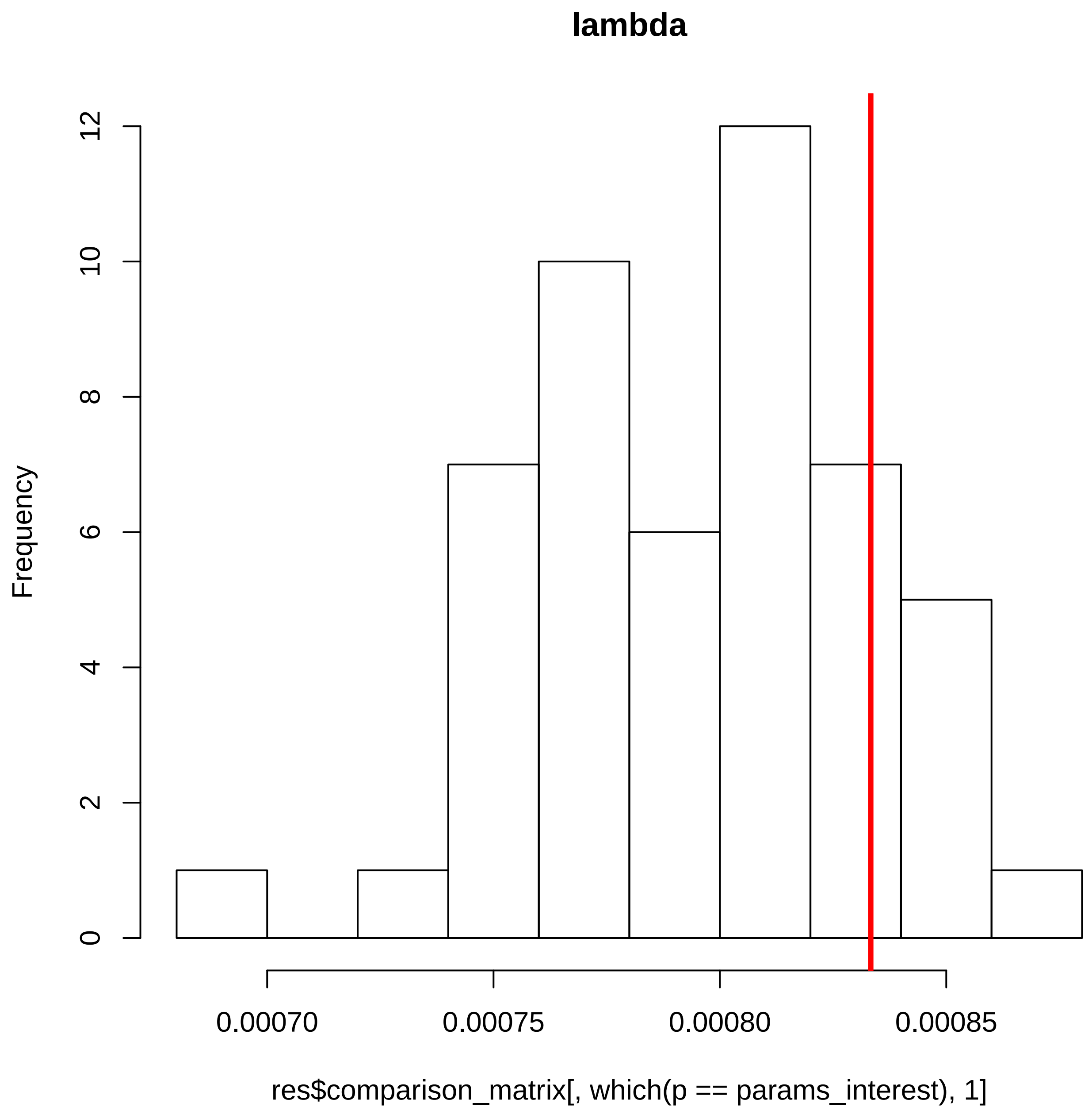




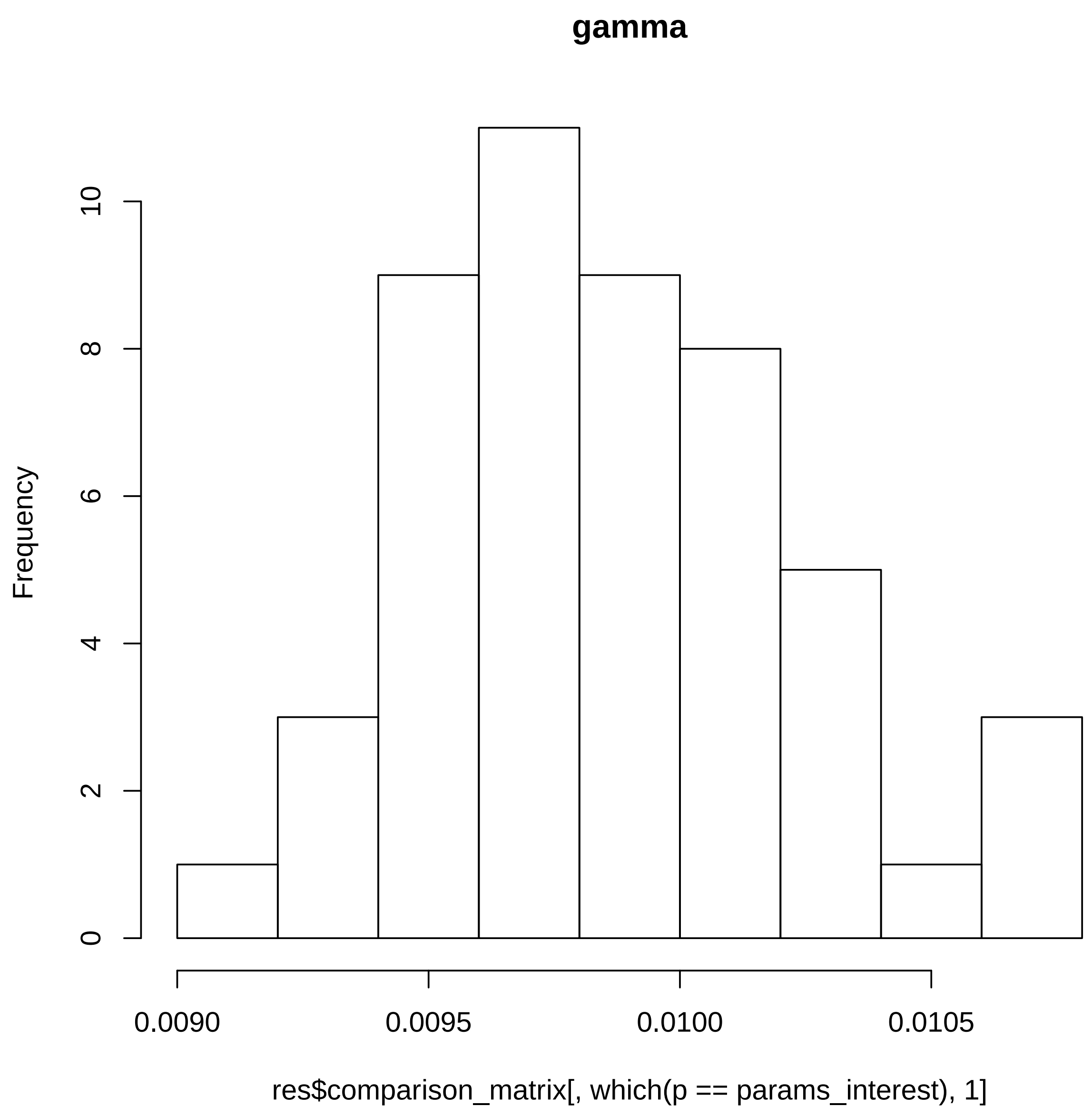
Run multiple random iterations to assess model fit and bias

```
for(p in params_interest){
  writeLines(sprintf('For parameter %s, %s%% are below the true parameter',p,100*sum(res$comparison_mat
                                                                                     unique(res$compa
  hist(res$comparison_matrix[,which(p==params_interest),1], main=p)
  abline(v=unique(res$comparison_matrix[,which(p==params_interest),2]),col='red',lwd=3)
}
```

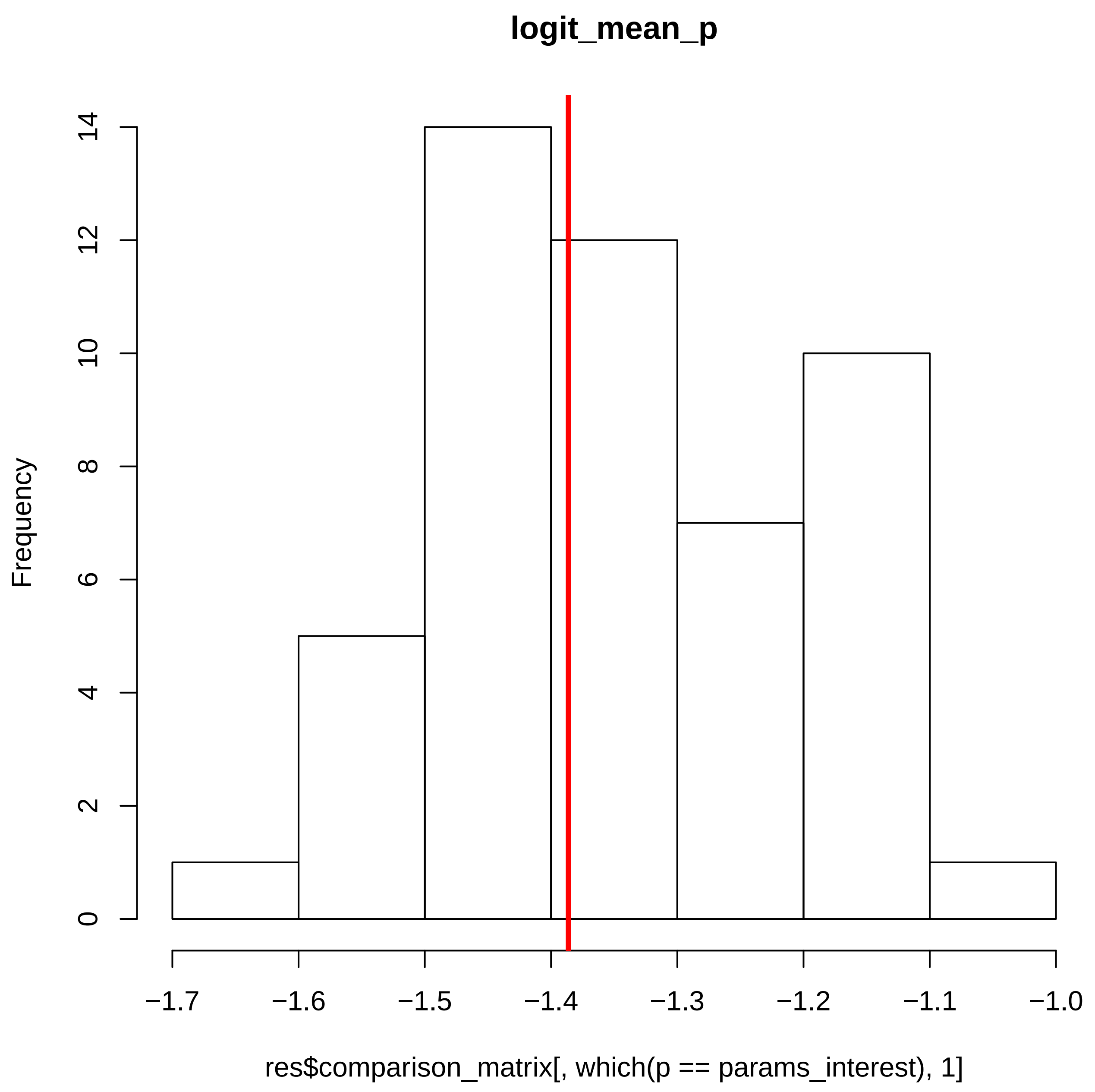
```
## For parameter lambda, 84% are below the true parameter
```



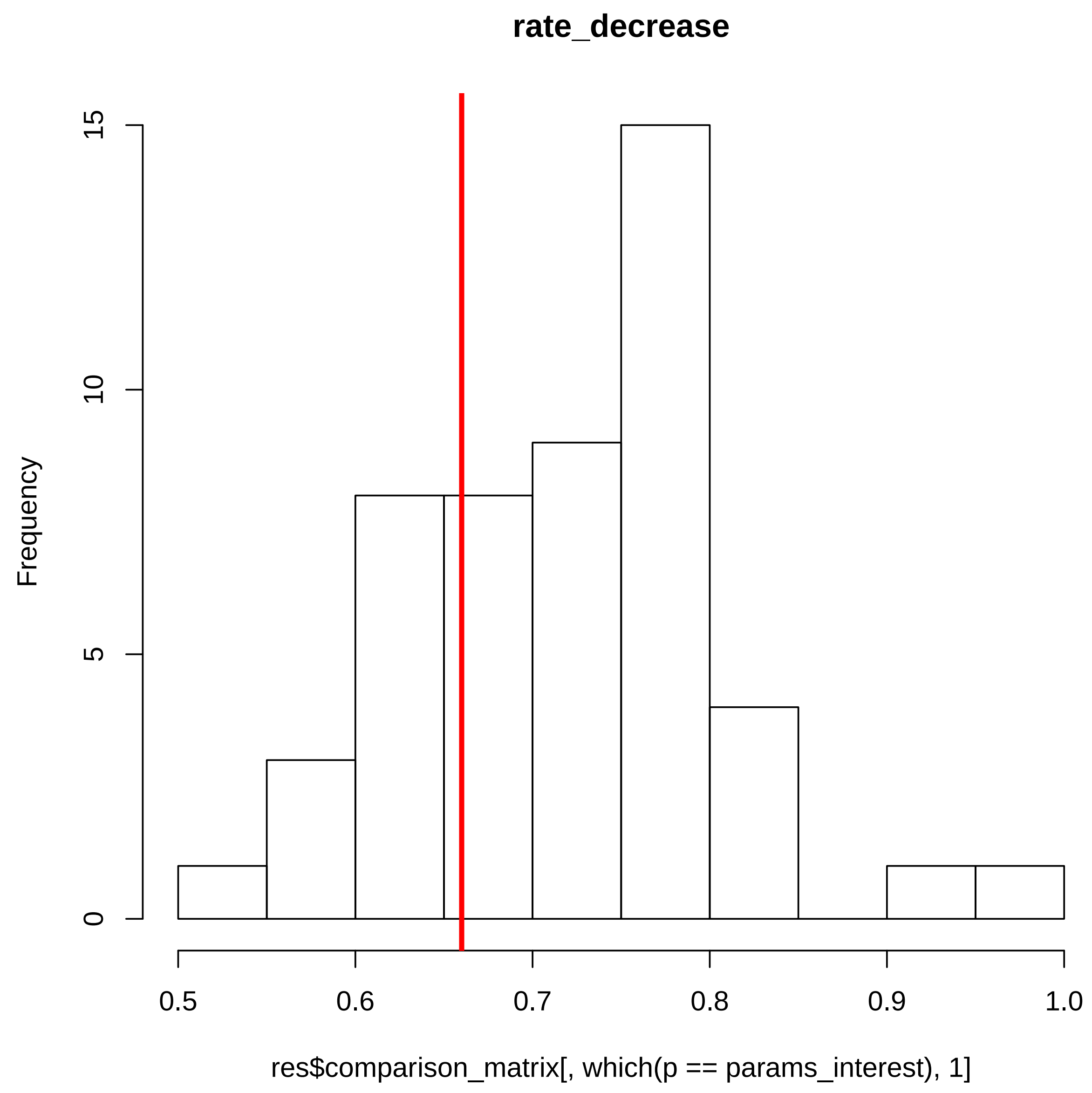
```
## For parameter gamma, 100% are below the true parameter
```



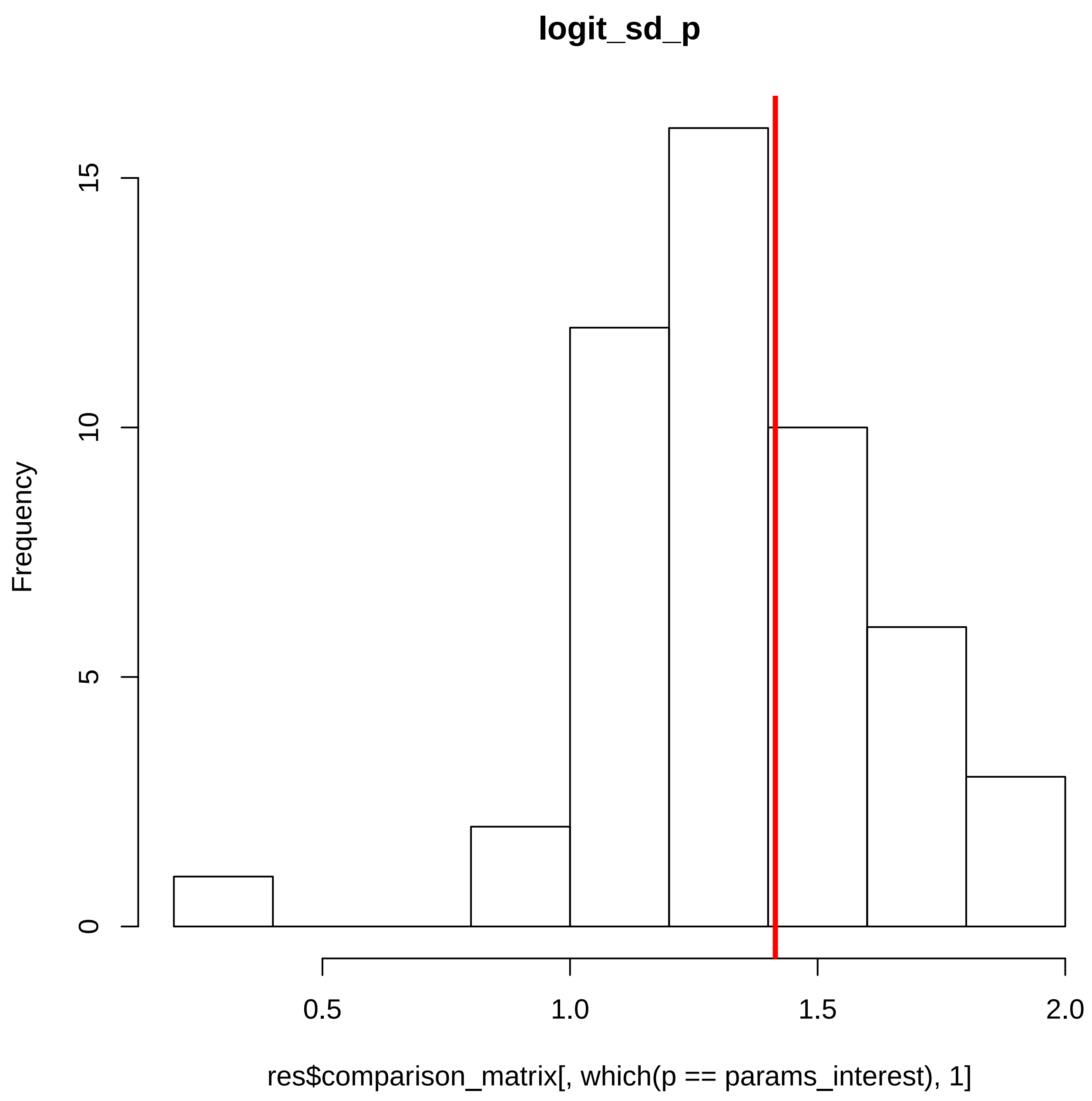
```
## For parameter logit_mean_p, 40% are below the true parameter
```



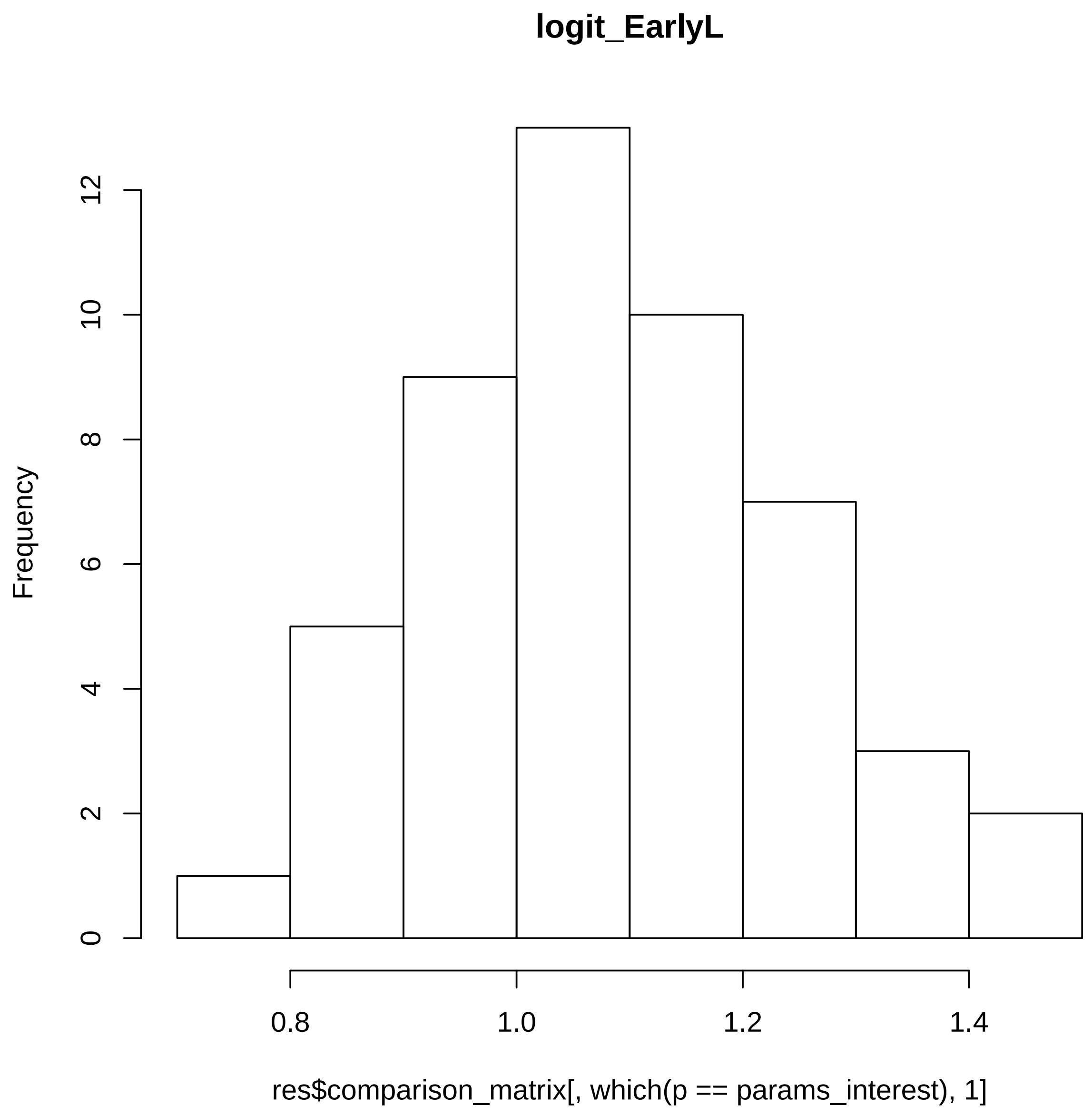
```
## For parameter rate_decrease, 26% are below the true parameter
```



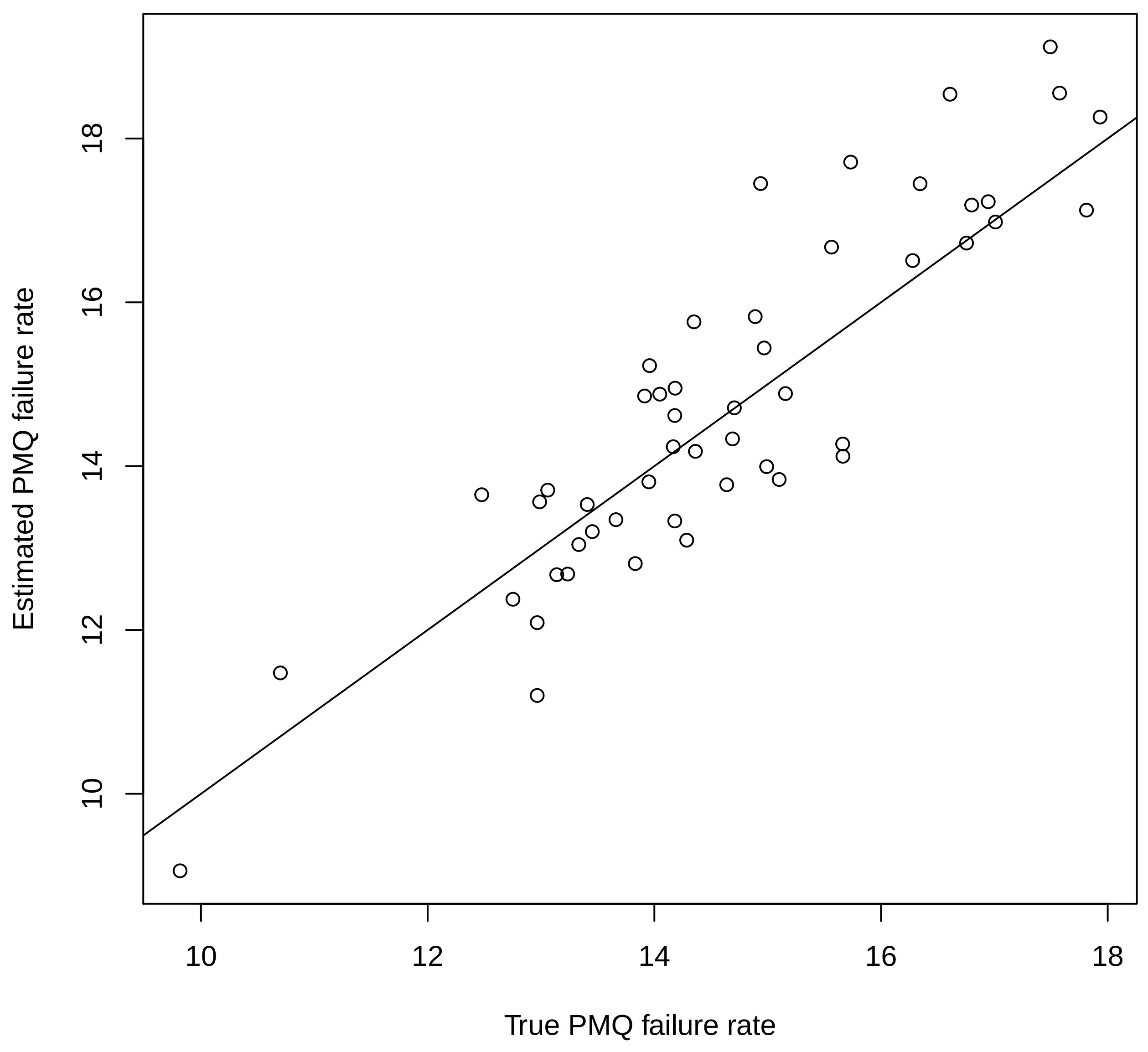
```
## For parameter logit_sd_p, 62% are below the true parameter
```



```
## For parameter logit_EarlyL, 0% are below the true parameter
```



```r
plot(100*res$failure_matrix[,1],100*res$failure_matrix[,2],
     xlab='True PMQ failure rate',ylab = 'Estimated PMQ failure rate')
lines(c(0,100),c(0,100))
```



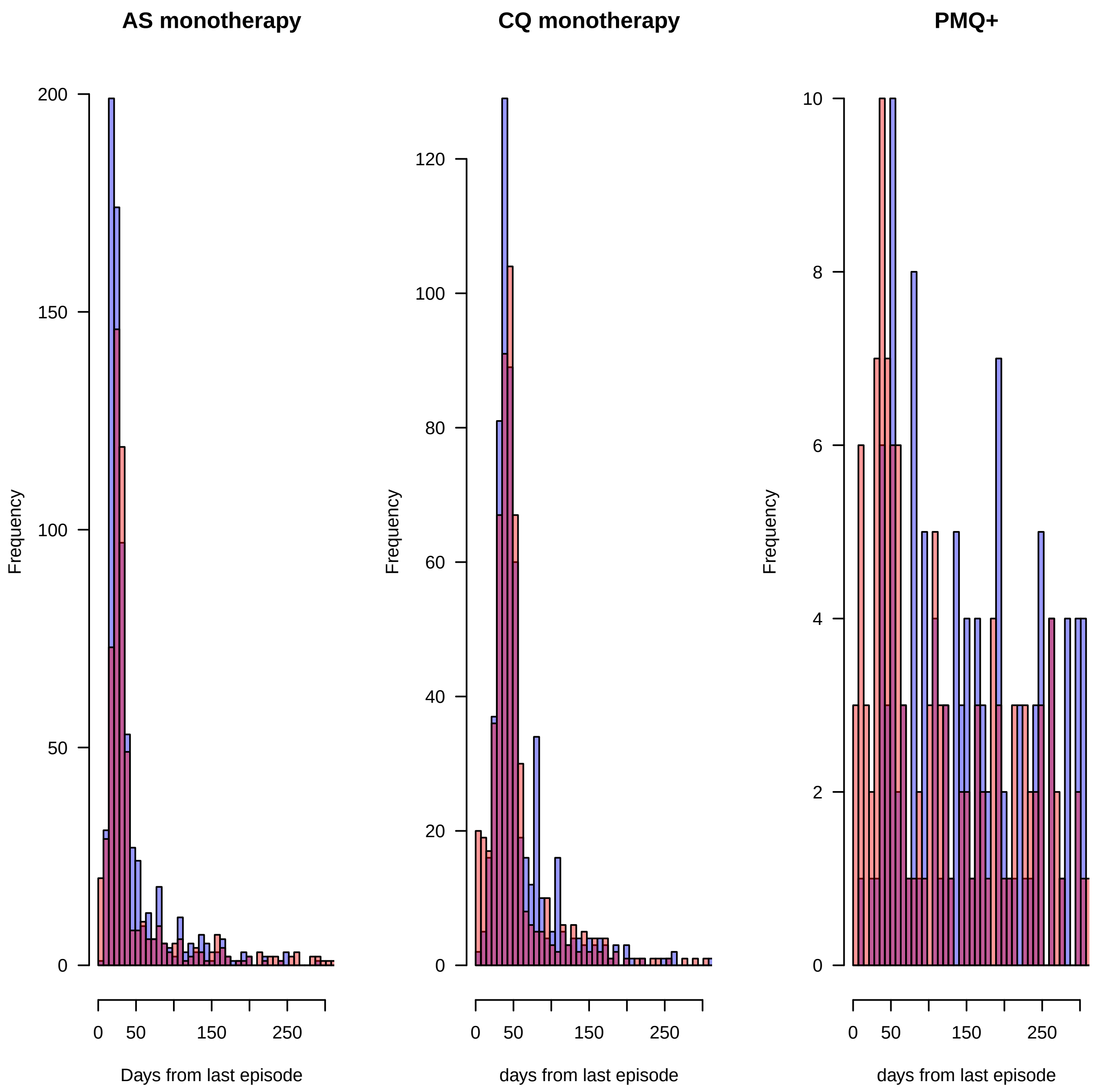
Posterior predictive p-values

In this section we simulate data from the posterior predictive for Model 2. We then apply a qualitative model check by comparing summary statistics from the posterior predicitive with the true data summary statistics.

```
load('../RData/TimingModel/Posterior_theta_samples.RData')
load('../RData/TimingModel/Combined_Time_Event.RData')


PMQ_FUP = Combined_Time_Data$FU_time[!duplicated(Combined_Time_Data$patientid)&
                                       Combined_Time_Data$arm_num=='CHQ/PMQ']
CQ_FUP = Combined_Time_Data$FU_time[!duplicated(Combined_Time_Data$patientid)&
                                      Combined_Time_Data$arm_num=='CHQ']
AS_FUP = Combined_Time_Data$FU_time[!duplicated(Combined_Time_Data$patientid)&
                                      Combined_Time_Data$arm_num=='AS']
N_PMQ = sum(Combined_Time_Data$arm_num=='CHQ/PMQ'&
```

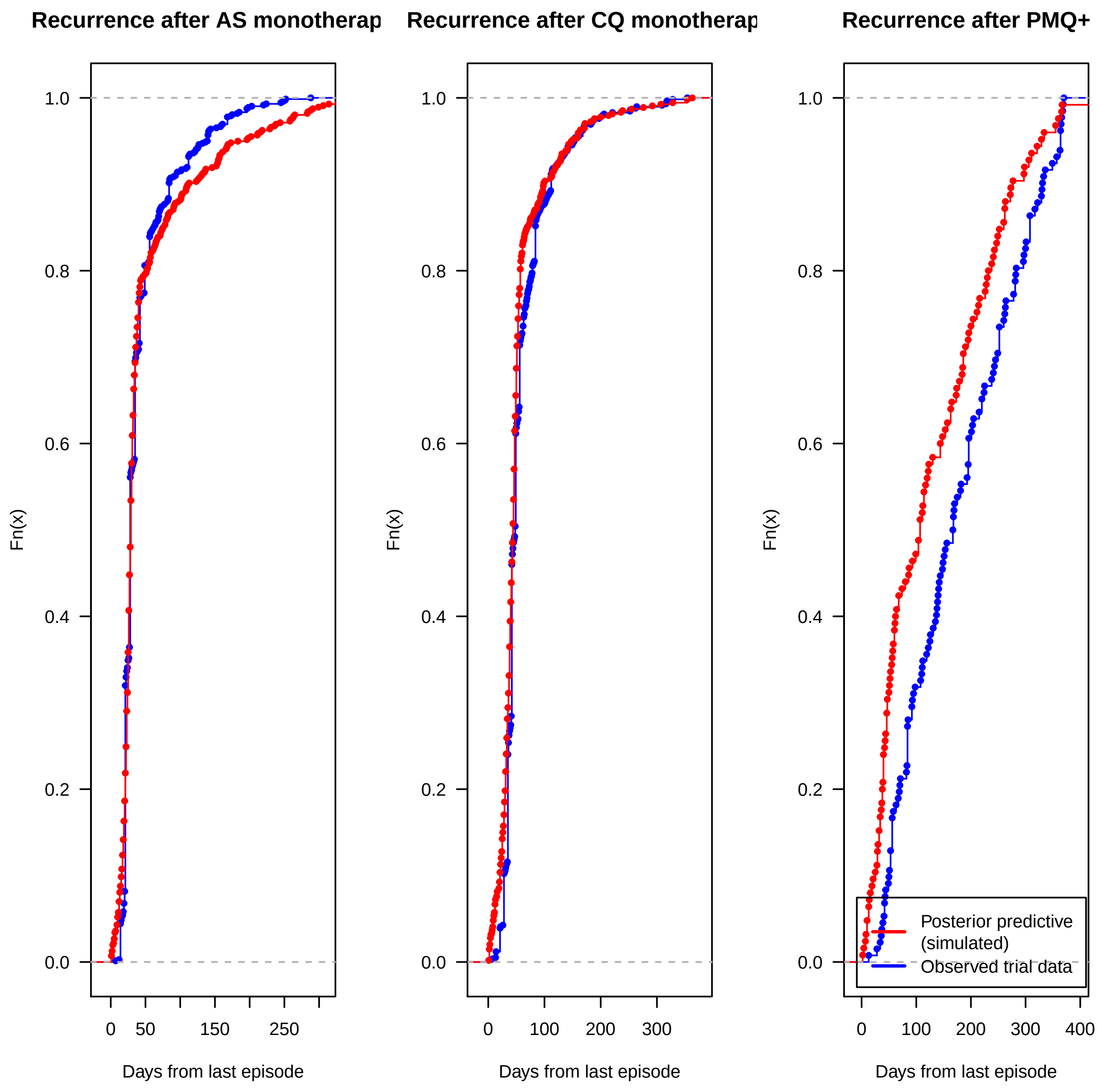
```r
               !duplicated(Combined_Time_Data$patientid))
N_CQ = sum(Combined_Time_Data$arm_num=='CHQ'&
              !duplicated(Combined_Time_Data$patientid))
N_AS = sum(Combined_Time_Data$arm_num=='AS'&
              !duplicated(Combined_Time_Data$patientid))
N_PMQ_BPD = sum(Combined_Time_Data$arm_num=='CHQ/PMQ'&
                  Combined_Time_Data$Study_Period==2 &
                  !duplicated(Combined_Time_Data$patientid))

set.seed(4543758)
params_M2_posterior = list()
i = sample(1:nrow(thetas2_matrix),1)
for(p in names(params_M2)){
  params_M2_posterior[[p]] = thetas2_matrix[i,p]
}

dat_posterior =
  simulate_dataset(N_PMQ = N_PMQ,N_CQ = N_CQ,N_AS = N_AS,
                   FUP_time = c(sample(x = PMQ_FUP,size = length(PMQ_FUP),replace = T),
                                sample(x = CQ_FUP,size = length(CQ_FUP),replace = T),
                                sample(x = AS_FUP,size = length(AS_FUP),replace = T)),
                   Study_Period = c(rep(2,N_PMQ_BPD),rep(2,N_PMQ-N_PMQ_BPD),rep(1,N_CQ+N_AS)),
                   data_generation_function = generate_patient_data_Model2,
                   params = params_M2_posterior)

AS_ind = dat_posterior$Simdata$Drug==0 &
  dat_posterior$Simdata$Censored==0
AS_ind_true = Combined_Time_Data$arm_num=='AS'&
  Combined_Time_Data$Censored==0
CQ_ind = dat_posterior$Simdata$Drug==1 &
  dat_posterior$Simdata$Censored==0
CQ_ind_true = Combined_Time_Data$arm_num=='CHQ'&
  Combined_Time_Data$Censored==0
PMQ_ind = dat_posterior$Simdata$Drug==2 & dat_posterior$Simdata$Censored==0
PMQ_ind_true = Combined_Time_Data$arm_num=='CHQ/PMQ'&
  Combined_Time_Data$Censored==0

par(las=1, mfrow=c(1,3))
#** Artesunate monotherapy: posterior predictive versus true data **
hist(Combined_Time_Data$Time_to_event[AS_ind_true],
     main = 'AS monotherapy',
     xlab='Days from last episode', col=alpha('blue',alpha = .4),
     breaks = seq(0,700,by=7), xlim = c(0,300))
hist(dat_posterior$Simdata$Durations[AS_ind],
     add=T, breaks = seq(0,700,by=7), col=alpha('red',alpha = .4))

#** Chloroquine monotherapy: posterior predictive versus true data **
hist(Combined_Time_Data$Time_to_event[CQ_ind_true], main = 'CQ monotherapy',
     xlab='days from last episode', col=alpha('blue',alpha = .4),
     breaks = seq(0,700,by=7), xlim = c(0,300))
hist(dat_posterior$Simdata$Durations[CQ_ind],
     add=T, breaks = seq(0,700,by=7), col=alpha('red',alpha = .4))
```

```
#** Primaquine+ : posterior predictive versus true data **
hist(Combined_Time_Data$Time_to_event[PMQ_ind_true], main = 'PMQ+',
     xlab='days from last episode', col=alpha('blue',alpha = .4),
     breaks = seq(0,700,by=7), xlim = c(0,300))
hist(dat_posterior$Simdata$Durations[PMQ_ind],
     add=T, breaks = seq(0,700,by=7), col=alpha('red',alpha = .4))
```



```
# *************** Empirical cdfs **************
par(las=1, mfrow=c(1,3))
plot.ecdf(Combined_Time_Data$Time_to_event[AS_ind_true],
          col='blue',verticals = T, xlab='Days from last episode',
          main='Recurrence after AS monotherapy')
plot.ecdf(dat_posterior$Simdata$Durations[AS_ind],
          add=T,col='red',verticals = T)
```

```r
plot.ecdf(Combined_Time_Data$Time_to_event[CQ_ind_true],
          col='blue',verticals = T, xlab='Days from last episode',
          main='Recurrence after CQ monotherapy')
plot.ecdf(dat_posterior$Simdata$Durations[CQ_ind],
          add=T,col='red',verticals = T)


plot.ecdf(Combined_Time_Data$Time_to_event[PMQ_ind_true],
          col='blue',verticals = T, xlab='Days from last episode',
          main='Recurrence after PMQ+')
plot.ecdf(dat_posterior$Simdata$Durations[PMQ_ind],
          add=T,col='red',verticals = T)
legend('bottomright',col=c('red','blue'),
       legend = c('Posterior predictive \n(simulated)',
                  'Observed trial data'),lwd=2, inset = 0.01)
```



```
Sim_Iterations = 200
res = array(dim = c(Sim_Iterations,3))
pb = txtProgressBar(min=1, max = Sim_Iterations,style = 3)
for(Sim in 1:Sim_Iterations){

  params_M2_posterior = list()
  i = sample(1:nrow(thetas2_matrix),1)
  for(p in names(params_M2)){
    params_M2_posterior[[p]] = thetas2_matrix[i,p]
  }
  dat_posterior =
    simulate_dataset(N_PMQ = N_PMQ,N_CQ = N_CQ,N_AS = N_AS,
                     FUP_time = c(sample(x = PMQ_FUP,size = length(PMQ_FUP),replace = T),
                                  sample(x = CQ_FUP,size = length(CQ_FUP),replace = T),
                                   sample(x = AS_FUP,size = length(AS_FUP),replace = T)),
                      Study_Period = c(rep(2,N_PMQ_BPD),rep(2,N_PMQ-N_PMQ_BPD),rep(1,N_CQ+N_AS)),
                      data_generation_function = generate_patient_data_Model2,
                      params = params_M2_posterior)

  AS_ind = dat_posterior$Simdata$Drug==0 &
    dat_posterior$Simdata$Censored==0
  CQ_ind = dat_posterior$Simdata$Drug==1 &
    dat_posterior$Simdata$Censored==0
  PMQ_ind = dat_posterior$Simdata$Drug==2 & dat_posterior$Simdata$Censored==0

  # compute recurrences per follow-up year
  res[Sim,1] = length(dat_posterior$Simdata$Durations[AS_ind])/
    (sum(dat_posterior$Simdata$Durations[dat_posterior$Simdata$Drug==0])/360)
  res[Sim,2] = length(dat_posterior$Simdata$Durations[CQ_ind])/
    (sum(dat_posterior$Simdata$Durations[dat_posterior$Simdata$Drug==1])/360)
  res[Sim,3] = length(dat_posterior$Simdata$Durations[PMQ_ind])/
    (sum(dat_posterior$Simdata$Durations[dat_posterior$Simdata$Drug==2])/360)
  setTxtProgressBar(pb,value = Sim)
}
```

```
##
  |
  |                                                                  |   0%
  |
  |                                                                  |   1%
  |
  |=                                                                 |   1%
  |
  |=                                                                 |   2%
  |
  |==                                                                |   3%
  |
  |==                                                                |   4%
  |
  |===                                                               |   4%
  |
  |===                                                               |   5%
  |
  |====                                                              |   6%
  |
  |====                                                              |   7%
  |
  |=====                                                             |   7%
  |
  |=====                                                             |   8%
  |
  |======                                                            |   9%
  |
  |======                                                            |  10%
  |
  |=======                                                           |  10%
  |
  |=======                                                           |  11%
```

```
|
|========                                                          |  12%
|
|========                                                          |  13%
|
|=========                                                         |  14%
|
|=========                                                         |  15%
|
|==========                                                        |  15%
|
|==========                                                        |  16%
|
|===========                                                       |  17%
|
|===========                                                       |  18%
|
|============                                                      |  18%
|
|============                                                      |  19%
|
|=============                                                     |  20%
|
|=============                                                     |  21%
|
|==============                                                    |  21%
|
|==============                                                    |  22%
|
|===============                                                   |  23%
|
|===============                                                   |  24%
|
|================                                                  |  24%
|
|================                                                  |  25%
|
|=================                                                 |  26%
|
|=================                                                 |  27%
|
|==================                                                |  27%
|
|==================                                                |  28%
|
|===================                                               |  29%
|
|===================                                               |  30%
|
|====================                                              |  30%
|
|====================                                              |  31%
|
|=====================                                             |  32%
```

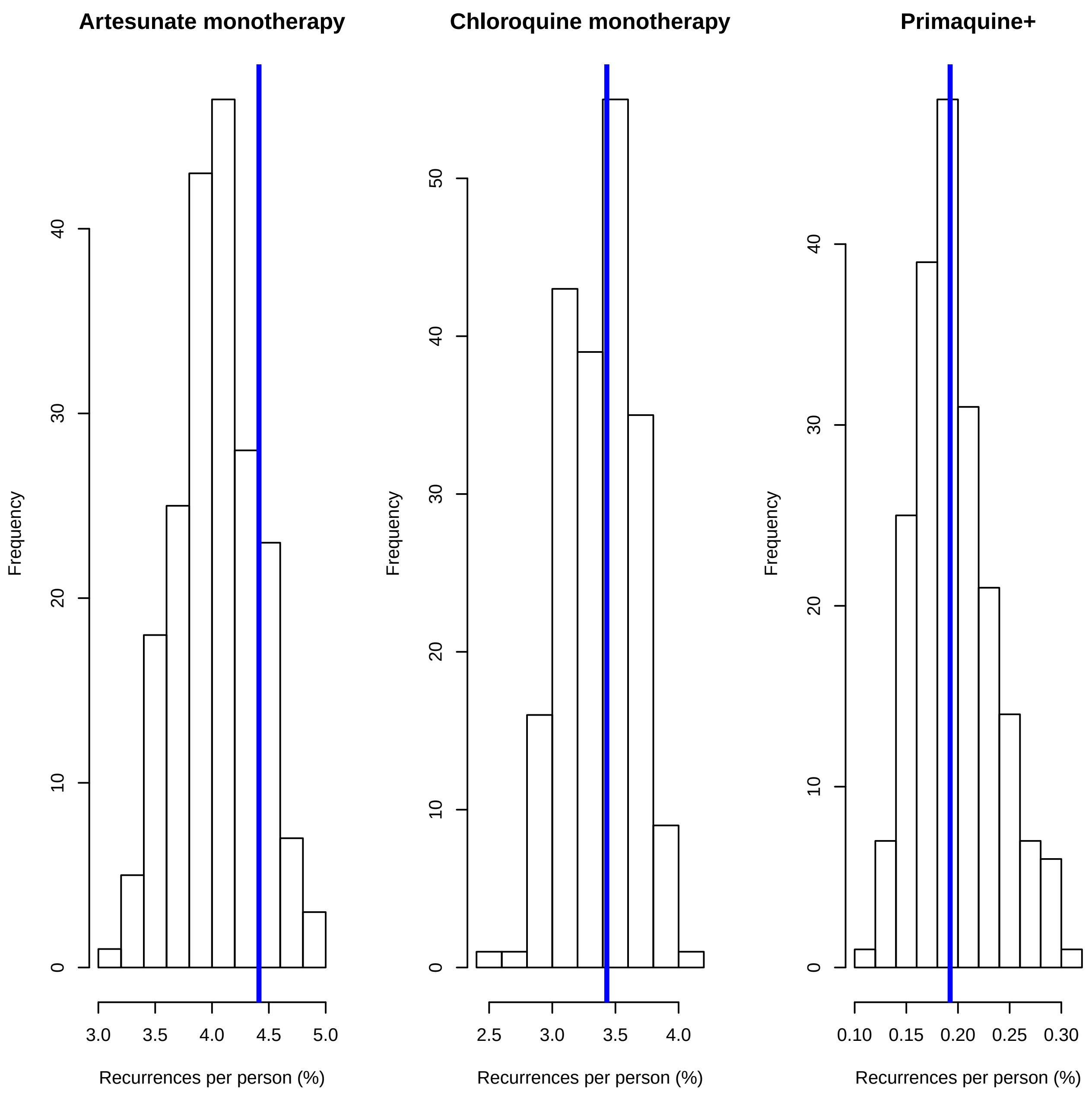
```
|
|=====================                                            |  33%
|
|======================                                           |  33%
|
|======================                                           |  34%
|
|=======================                                          |  35%
|
|=======================                                          |  36%
|
|========================                                         |  36%
|
|========================                                         |  37%
|
|========================                                         |  38%
|
|=========================                                        |  38%
|
|=========================                                        |  39%
|
|==========================                                       |  40%
|
|==========================                                       |  41%
|
|===========================                                      |  41%
|
|===========================                                      |  42%
|
|============================                                     |  43%
|
|============================                                     |  44%
|
|=============================                                    |  44%
|
|=============================                                    |  45%
|
|==============================                                   |  46%
|
|==============================                                   |  47%
|
|===============================                                  |  47%
|
|===============================                                  |  48%
|
|================================                                 |  49%
|
|================================                                 |  50%
|
|=================================                                |  50%
|
|=================================                                |  51%
|
|==================================                               |  52%
```

```
  |
  |==================================                               |  53%
  |
  |===================================                              |  53%
  |
  |===================================                              |  54%
  |
  |====================================                             |  55%
  |
  |====================================                             |  56%
  |
  |=====================================                            |  56%
  |
  |=====================================                            |  57%
  |
  |======================================                           |  58%
  |
  |======================================                           |  59%
  |
  |=======================================                          |  59%
  |
  |=======================================                          |  60%
  |
  |========================================                         |  61%
  |
  |========================================                         |  62%
  |
  |=========================================                        |  62%
  |
  |=========================================                        |  63%
  |
  |=========================================                        |  64%
  |
  |==========================================                       |  64%
  |
  |==========================================                       |  65%
  |
  |===========================================                      |  66%
  |
  |===========================================                      |  67%
  |
  |============================================                     |  67%
  |
  |============================================                     |  68%
  |
  |=============================================                    |  69%
  |
  |=============================================                    |  70%
  |
  |==============================================                   |  70%
  |
  |==============================================                   |  71%
  |
  |===============================================                  |  72%
```

```
|
|===============================================                  |  73%
|
|================================================                 |  73%
|
|================================================                 |  74%
|
|=================================================                |  75%
|
|=================================================                |  76%
|
|==================================================               |  76%
|
|==================================================               |  77%
|
|===================================================              |  78%
|
|===================================================              |  79%
|
|====================================================             |  79%
|
|====================================================             |  80%
|
|=====================================================            |  81%
|
|=====================================================            |  82%
|
|======================================================           |  82%
|
|======================================================           |  83%
|
|=======================================================          |  84%
|
|=======================================================          |  85%
|
|========================================================         |  85%
|
|========================================================         |  86%
|
|=========================================================        |  87%
|
|=========================================================        |  88%
|
|==========================================================       |  89%
|
|==========================================================       |  90%
|
|===========================================================      |  90%
|
|===========================================================      |  91%
|
|============================================================     |  92%
|
|============================================================     |  93%
```

```
  |
  |=============================================================    |  93%
  |
  |=============================================================    |  94%
  |
  |==============================================================   |  95%
  |
  |==============================================================   |  96%
  |
  |===============================================================  |  96%
  |
  |===============================================================  |  97%
  |
  |================================================================ |  98%
  |
  |================================================================ |  99%
  |
  |=================================================================|  99%
  |
  |=================================================================| 100%
```

```r
par(mfrow=c(1,3))
hist(res[,1], xlab = 'Recurrences per person (%)', main = 'Artesunate monotherapy')
true_stat_AS = length(Combined_Time_Data$Time_to_event[AS_ind_true])/
  (sum(Combined_Time_Data$Time_to_event[Combined_Time_Data$arm_num=='AS'])/360)
abline(v = true_stat_AS, col='blue',lwd=3)

hist(res[,2], xlab = 'Recurrences per person (%)', main = 'Chloroquine monotherapy')
true_stat_CQ = length(Combined_Time_Data$Time_to_event[CQ_ind_true])/
  (sum(Combined_Time_Data$Time_to_event[Combined_Time_Data$arm_num=='CHQ'])/360)
abline(v = true_stat_CQ,  col='blue',lwd=3)

hist(res[,3], xlab = 'Recurrences per person (%)', main = 'Primaquine+')
true_stat_PMQ = length(Combined_Time_Data$Time_to_event[PMQ_ind_true])/
  (sum(Combined_Time_Data$Time_to_event[Combined_Time_Data$arm_num=='CHQ/PMQ'])/360)
abline(v = true_stat_PMQ,col='blue',lwd=3)
```



```
PP_pval_AS = 1-sum(res[,1]>true_stat_AS)/Sim_Iterations
writeLines(sprintf('The posterior predictive p value for the number of recurrences per person in the AS
```

```
## The posterior predictive p value for the number of recurrences per person in the AS arm is 0.845
```

```
PP_pval_CQ = 1-sum(res[,2]>true_stat_CQ)/Sim_Iterations
writeLines(sprintf('The posterior predictive p value for the number of recurrences per person in the CQ
```

```
## The posterior predictive p value for the number of recurrences per person in the CQ arm is 0.525
```

```
PP_pval_PMQ = 1-sum(res[,3]>true_stat_PMQ)/Sim_Iterations
writeLines(sprintf('The posterior predictive p value for the number of recurrences per person in the PMQ
```

```
## The posterior predictive p value for the number of recurrences per person in the PMQ+ arm is 0.535
```