Timing model using stan

1 Preliminaries

Load R packages.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: StanHeaders

## rstan (Version 2.17.3, 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)
```

2 Statistical Models

2.1 Notationa and definition of terms

The i^{th} time to recurrent infection is denoted T_i (initially we ignore clustering by individual). This is the duration in days since the previous infection was treated. The drug that corresponds to the timing of that infection (i.e. the drug that was use to treat the previous infection) is denoted D_i.

For recurrent vivax infections we use the following terms:

- A reLapse (L) is a recurent infection derived from a 'waking' hypnozoite. This can be genetically related to the previous infection (clone or sibling: homologous reLapse), or it can be different (heterologous reLapse).
- A reInfection (I) is a recurrent infection derived from a new mosquito bite. We assume that this will be heterologous (genetically different) from prvious infections, unless there is a clonal outbreak.
- A reCrudescence (C) is the same blood stream infection as the previous infection, which was not completely cleared by the blood stream drug.

2.2 Model structures

This code implements four different 'structural' mixture models:

- Model 0: Time-to-recurrence in individuals who did not receive radical cure is modelled as a mixture of two exponential distributions (one is the reLapse component, the other is the reInfection component). Time-to-recurrence in primaquine treated individuals is modelled as a single exponential (only the reinfection component, this assumes 100% efficacy of primaquine).

$$T_i \sim \begin{cases} pE(\lambda) + (1-p)E(\gamma) & \text{if } D_i \text{ is AS or CQ} \\ E(\lambda) & \text{if } D_i \text{ is CQ+PMQ} \end{cases}$$

where $E(\lambda)$ is an exponential distribution with rate parameter λ and p is the mixing proportion between reInfections and reLapses.

- Model 1: Time-to-recurrence in individuals who did not receive radical cure is modelled as a mixture between a Weibull distribution (reLapse component) and an exponential distribution (reInfection) component. Time-to-recurrence in primaquine treated individuals is modelled with the same exponential distribution (also assuming 100% efficacy of primaquine).

$$T_i \sim \begin{cases} pE(\lambda) + (1-p)W(\mu_{D_i}, \sigma_{D_i}) & \text{if } D_i \text{ is AS or CQ} \\ E(\lambda) & \text{if } D_i \text{ is CQ+PMQ} \end{cases}$$

where $W(\mu_{D_i}, \sigma_{D_i})$ is a Weibull distribution with scale parameter μ_{D_i} and shape parameter σ_{D_i}.

- Model 2: Time-to-recurrence in individuals who did not receive radical cure is modelled as a mixture between a Weibull distribution (reLapse component 1: early/highly periodic reLapse), an exponential distribution with short mean occurence (reLapse component 2: late/random reLapse), and an exponential distribution with long mean occurence (reInfection component). Time-to-recurrence in primaquine treated individuals is modelled with just the same long mean exponential distribution (also assumes 100% efficacy of primaquine).

$$T_i \sim \begin{cases} pE(\lambda) + (1-p)\left\{qW(\mu_{D_i}, \sigma_{D_i}) + (1-q)E(\gamma)\right\} & \text{if } D_i \text{ is AS or CQ} \\ E(\lambda) & \text{if } D_i \text{ is CQ+PMQ} \end{cases}$$

where q is the mixing proportion between 'periodic' and 'random' reLapses. The 'random' reLapses are parametrised by the rate constant γ.

- Model 3: Time-to-recurrence in all individuals (primaquine treatment or not) is modelled with the same 3-component mixture as in Model 2. However the mixing proportions between reLapse and reInfection differ between the primaquine and non primaquine groups (not assuming 100% efficacy).

$$T_i \sim \begin{cases} p_1E(\lambda) + (1-p_1)\left\{qW(\mu_{D_i}, \sigma_{D_i}) + (1-q)E(\gamma)\right\} & \text{if } D_i = \text{AS or CQ} \\ p_2E(\lambda) + (1-p_2)\left\{qW(\mu_{D_i}, \sigma_{D_i}) + (1-q)E(\gamma)\right\} & \text{if } D_i = \text{CQ+PMQ} \end{cases}$$

3 Stan code

3.1 Model 0

This is the fully null model. ReLapses occur at 'random': e.g. they can be described by an exponential distribution. ReInfections also occur at 'random' but with a different rate parameter.

3.2 Model 1

3.3 Model 2: Mixture of three components

3.4 Model 3: Mixture of three components and reLapses after PMQ

4 Data

```
# Load the dataset - already been partially cleaned
load('../../RData/VHX_ClinicalData.RData')
# Get rid of the very short durations
vhx_dat = filter(vhx_dat, Time_to_event>10)
#vhx_dat = filter(vhx_dat, arm_num=='CHQ')
# We sort the ids so that ids go from 1...Nmax
ids = unique(vhx_dat$patientid)
rank_ids = 1:length(ids)
vhx_dat$ordered_ids = unlist(sapply(1:length(ids),
                                    function(x) rep(rank_ids[x],
                                                    sum(vhx_dat$patientid==ids[x])))
)
# Turn drug into a numeric vector
numeric_drug = as.integer(revalue(vhx_dat$arm_num, c('AS'='0','CHQ'='1','CHQ/PMQ'='2')))
```

5 Prior specification

```
# The hierachical parameters defining the prior distributions for model 1
Prior_params_M0 = list(mu_inv_lambda = 600,
                       sigma_inv_lambda = 25,
                       p_a1 = 2,
                       p_a2 = 19,
                       mu_inv_gamma = 40,
                       sigma_inv_gamma = 2)
# The hierachical parameters defining the prior distributions for model 1
Prior_params_M1 = list(mu_inv_lambda = 600,
                       sigma_inv_lambda = 50,
                       p_a1 = 1,
                       p_a2 = 9,
                       mu_AS_shape = 2,
                       sigma_AS_shape = 1,
                       mu_AS_scale = 25,
                       sigma_AS_scale = 5,
                       mu_CQ_shape = 2,
                       sigma_CQ_shape = 1,
                       mu_CQ_scale = 42,
                       sigma_CQ_scale = 5)
# Model 2 has the same parameters with a few extra
Prior_params_M2 = c(Prior_params_M1,
                    Early_L_a1 = 6,
                    Early_L_a2 = 4,
                    mu_inv_gamma = 100,
```

```
                    sigma_inv_gamma = 25)
# Model 3: extra parameters
Prior_params_M3 = c(Prior_params_M2,
                    p_PMQ_a1 = 9,
                    p_PMQ_a2 = 1)
```

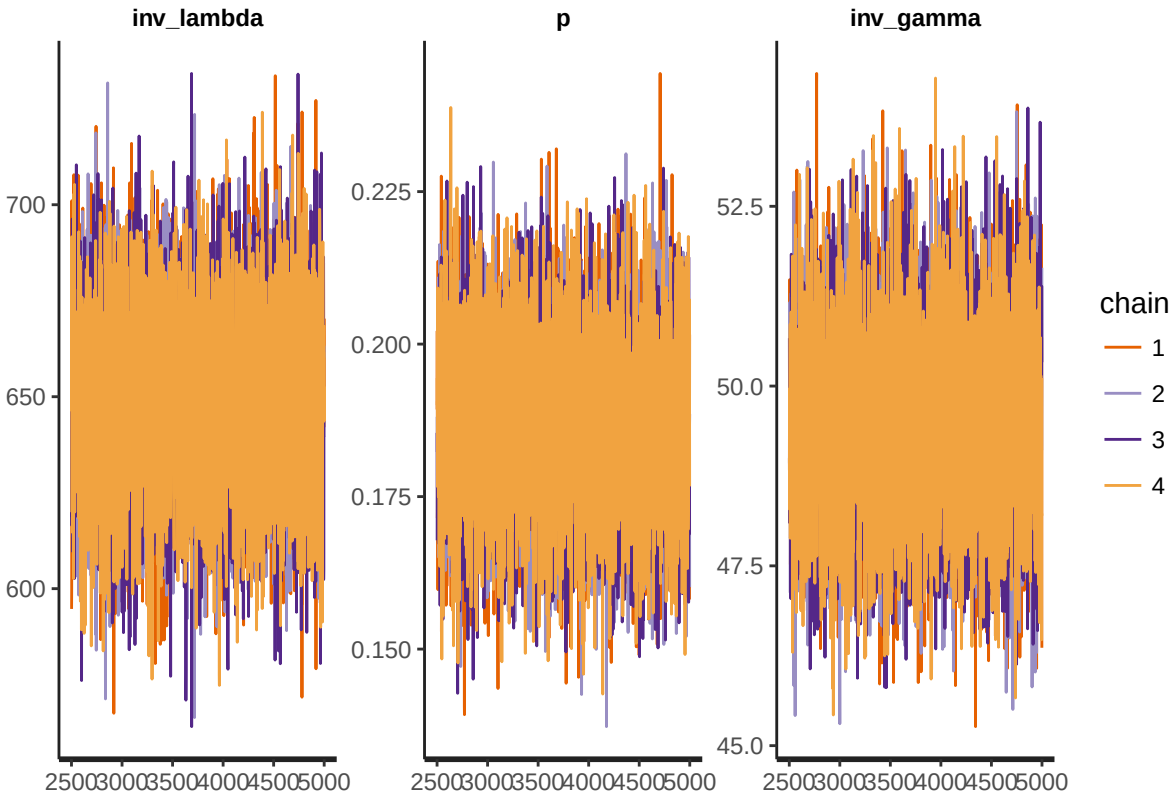
6 Run Model

Run with X parallel chains. This depends on local computing power.

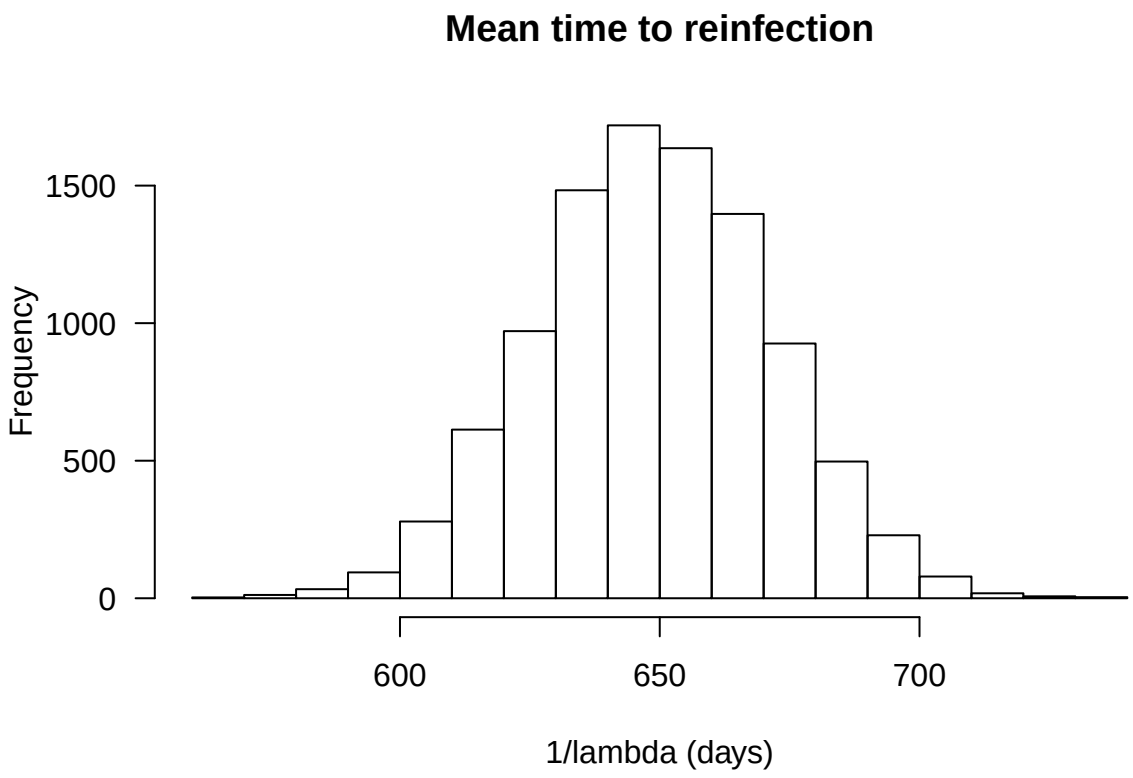
7 Plot output

7.1 Model 0

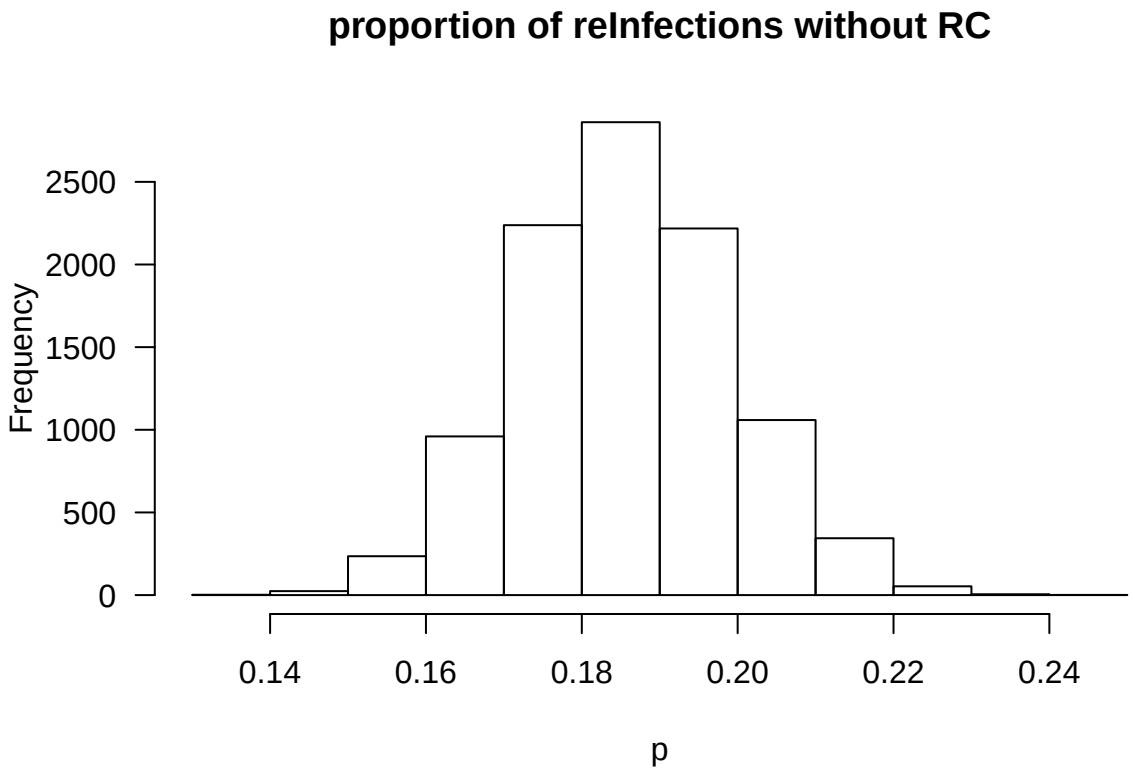
```
par(las=1)
traceplot(mod0, c('inv_lambda','p','inv_gamma'))
```




```
thetas = extract(mod0)
hist(thetas$inv_lambda, main='Mean time to reinfection',xlab='1/lambda (days)')
```

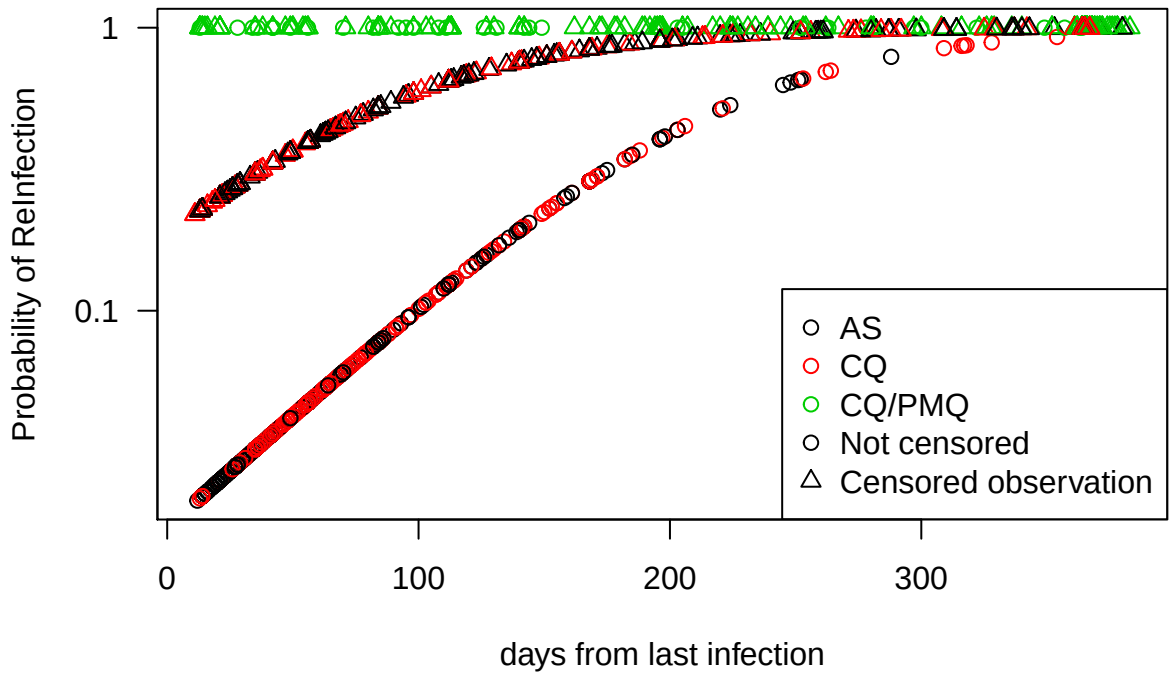



```
hist(thetas$p, main = 'proportion of reInfections without RC', xlab='p')
```



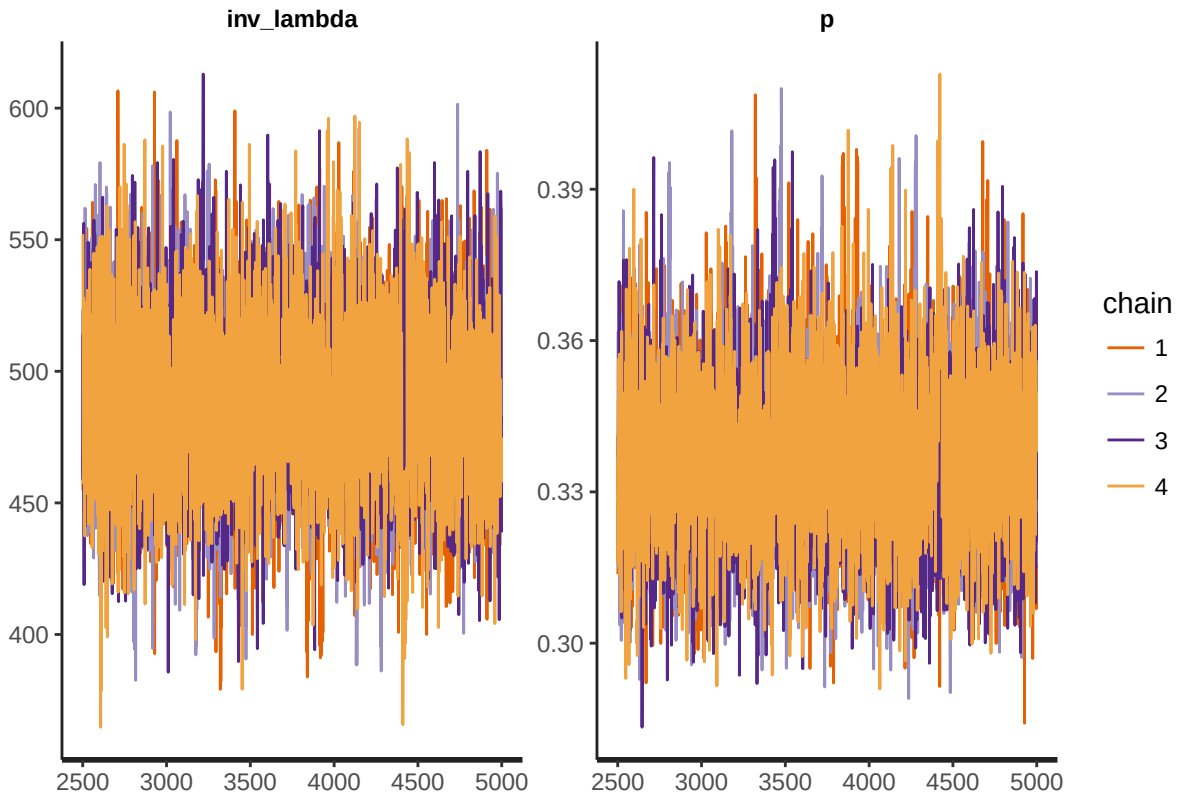

```
labels = extract(mod0, 'prob_labels')$prob_labels
mean_labels = apply(labels, c(2,3), mean)
plot(vhx_dat$Time_to_event, log10(mean_labels[,1]),
```

```
     col = numeric_drug+1,
     pch = as.numeric(vhx_dat$Right_Truncated)+1,
     ylab='Probability of ReInfection',
     xlab='days from last infection',yaxt='n')
axis(2, at= -2:0, labels= 10^(-2:0))
legend('bottomright',
       legend = c('AS','CQ','CQ/PMQ',
                  'Not censored','Censored observation'),
       col=c(1:3,1,1),pch = c(rep(1,3),1:2))
```

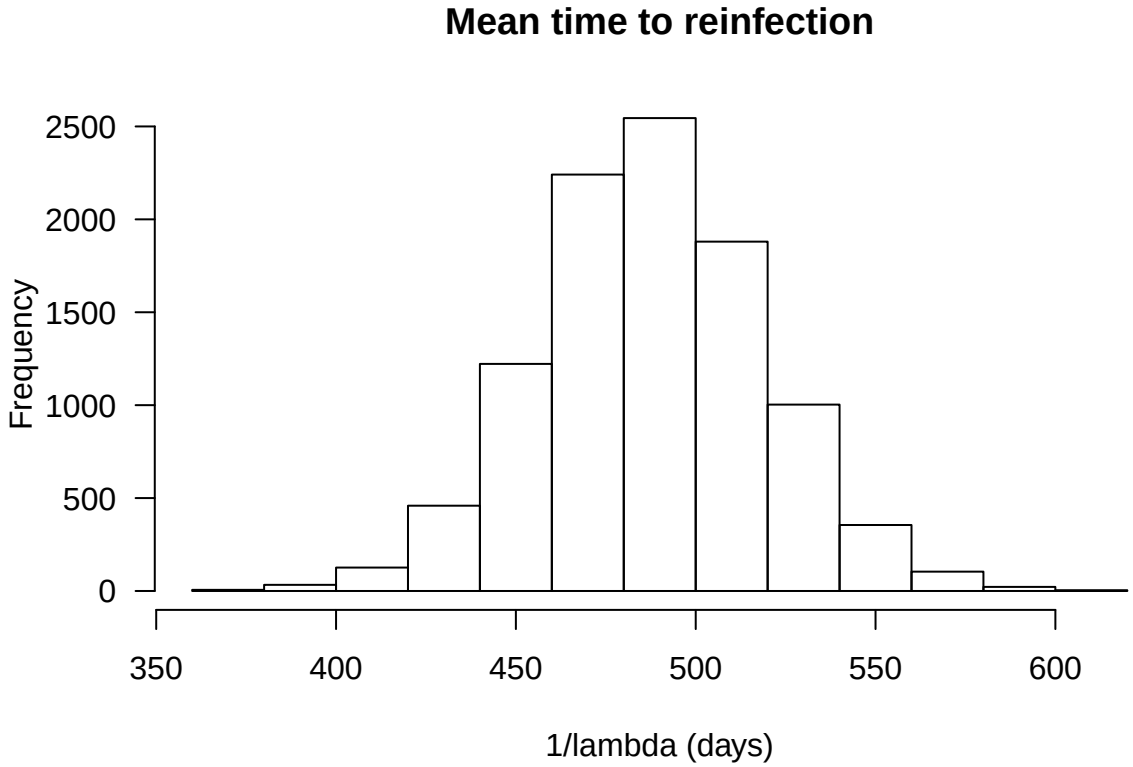


7.2 Model 1

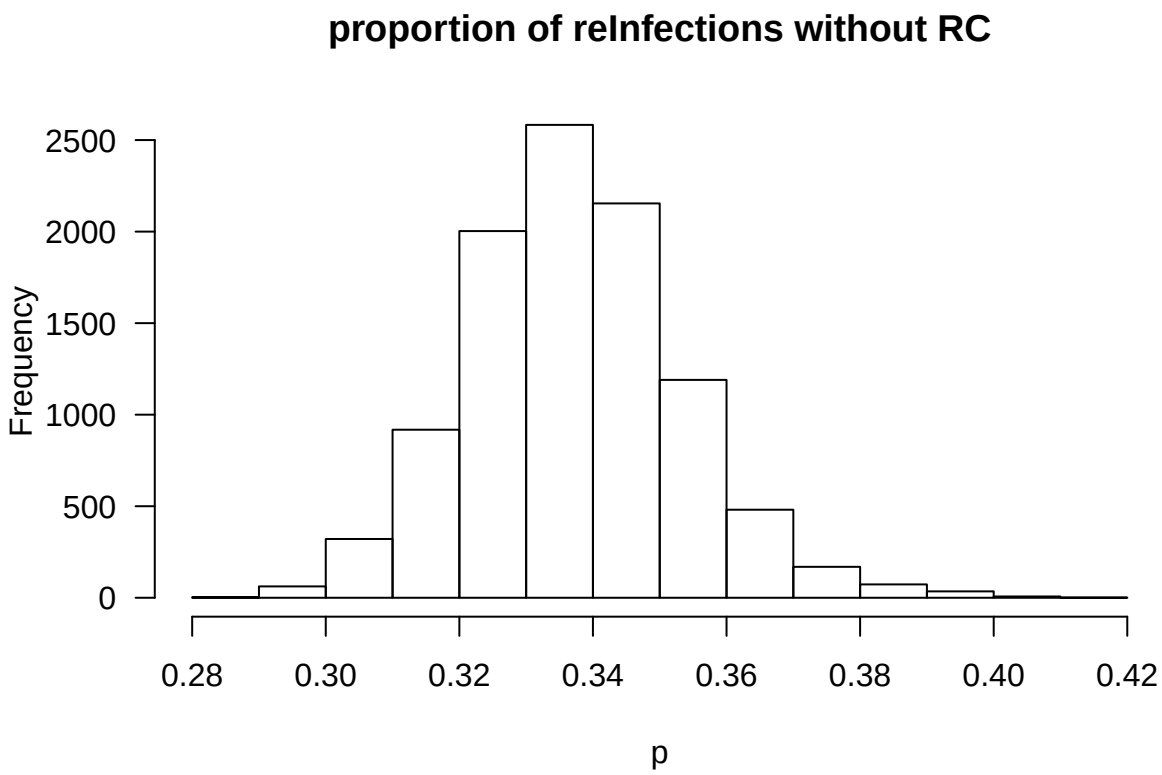
```
par(las=1)
traceplot(mod1, c('inv_lambda','p'))
```



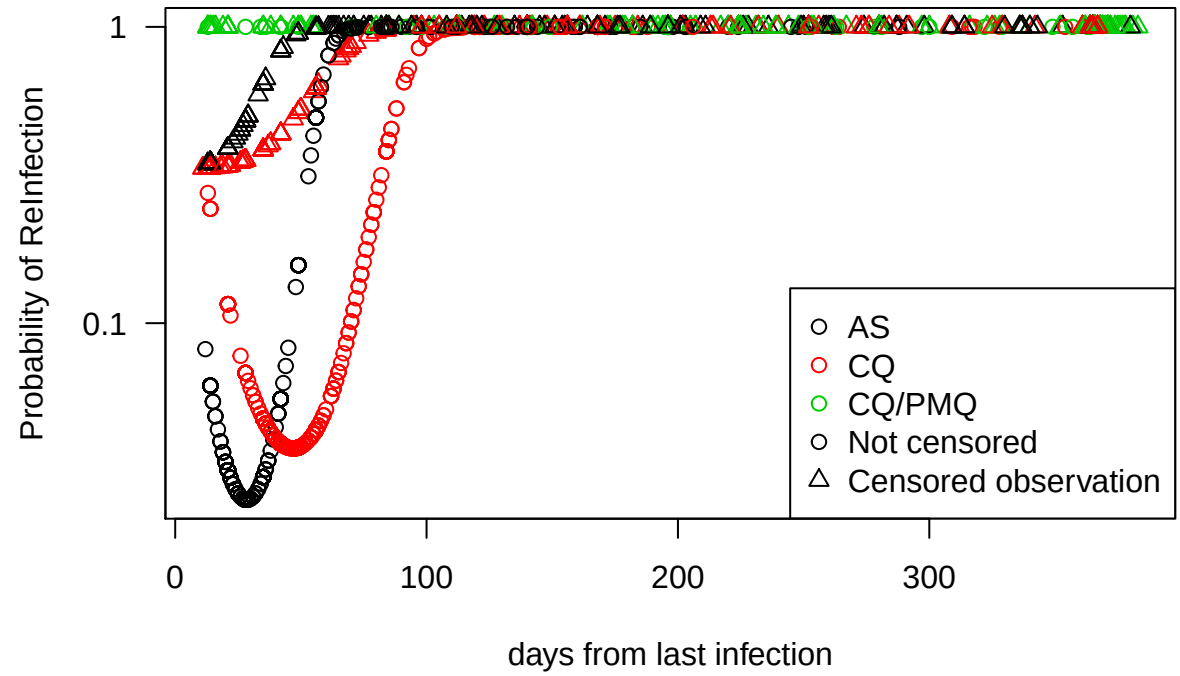

```
thetas = extract(mod1)
hist(thetas$inv_lambda, main='Mean time to reinfection',xlab='1/lambda (days)')
```



```r
hist(thetas$p, main = 'proportion of reInfections without RC', xlab='p')
```

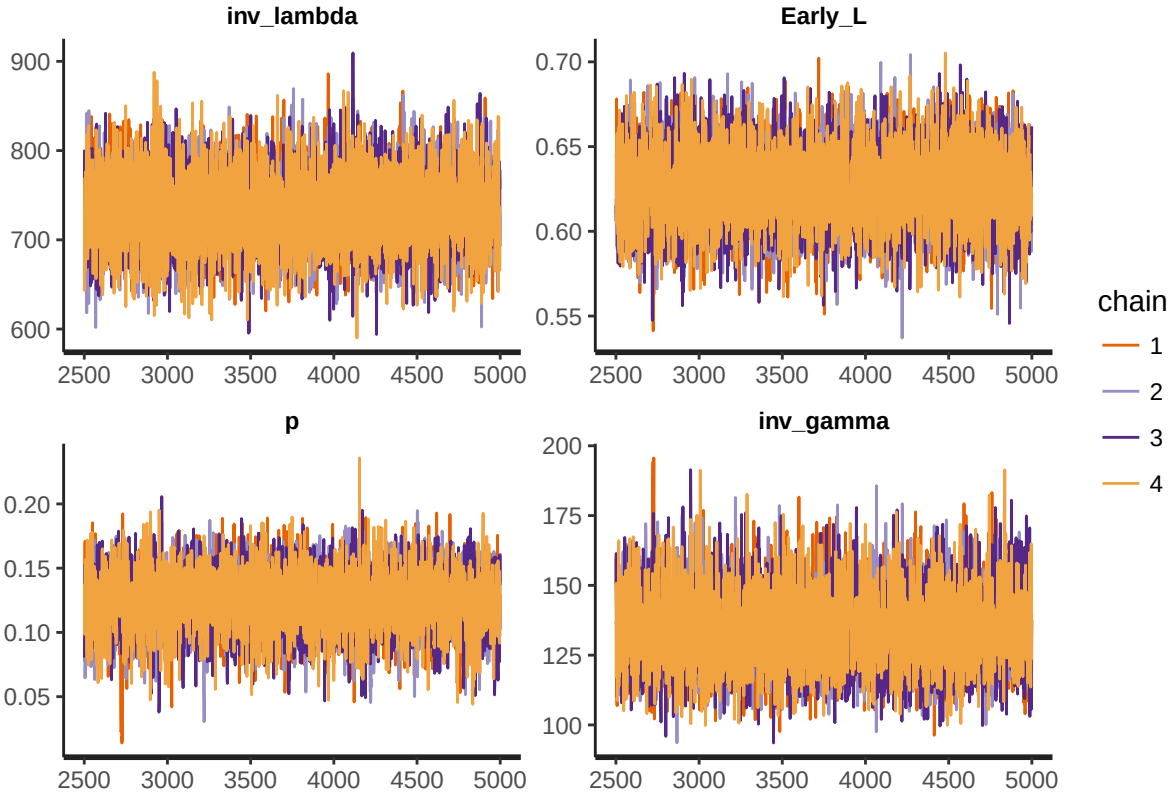



```r
labels = extract(mod1, 'prob_labels')$prob_labels
mean_labels = apply(labels, c(2,3), mean)
plot(vhx_dat$Time_to_event, log10(mean_labels[,1]), col = numeric_drug+1, pch = as.numeric(vhx_dat$Righ
axis(2, at= -2:0, labels= 10^(-2:0))
legend('bottomright',legend = c('AS','CQ','CQ/PMQ', 'Not censored','Censored observation'), col=c(1:3,1
```

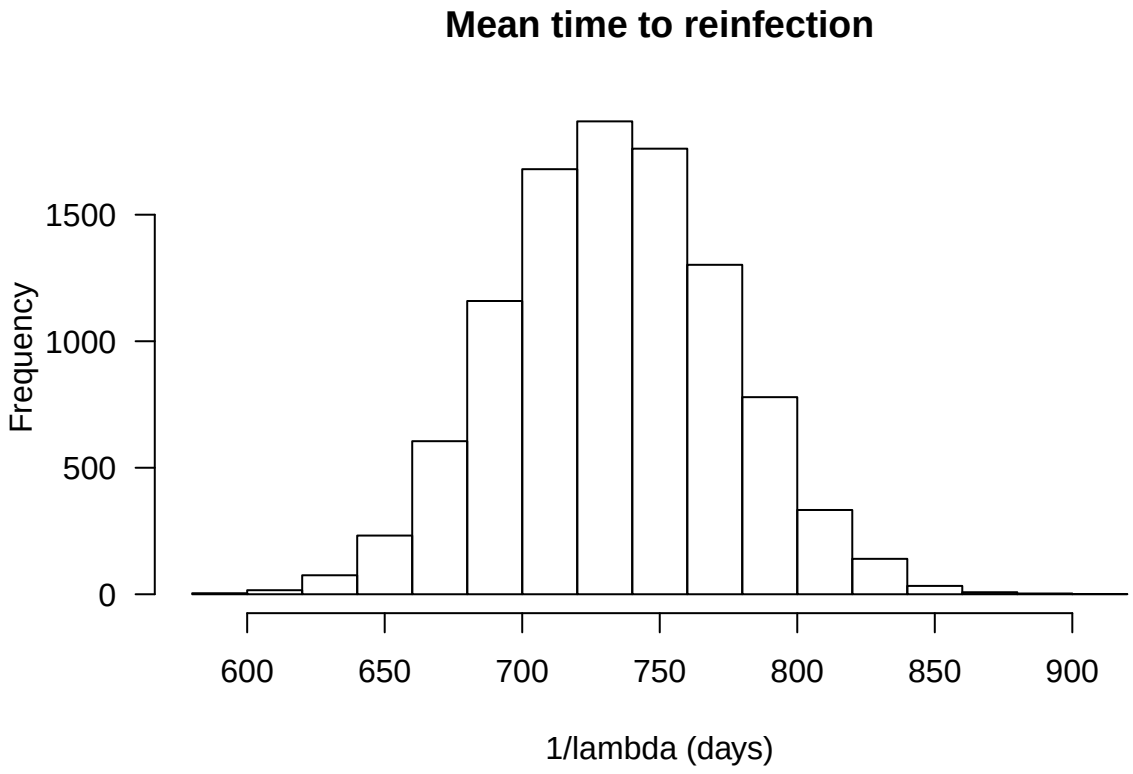


7.3 Model 2

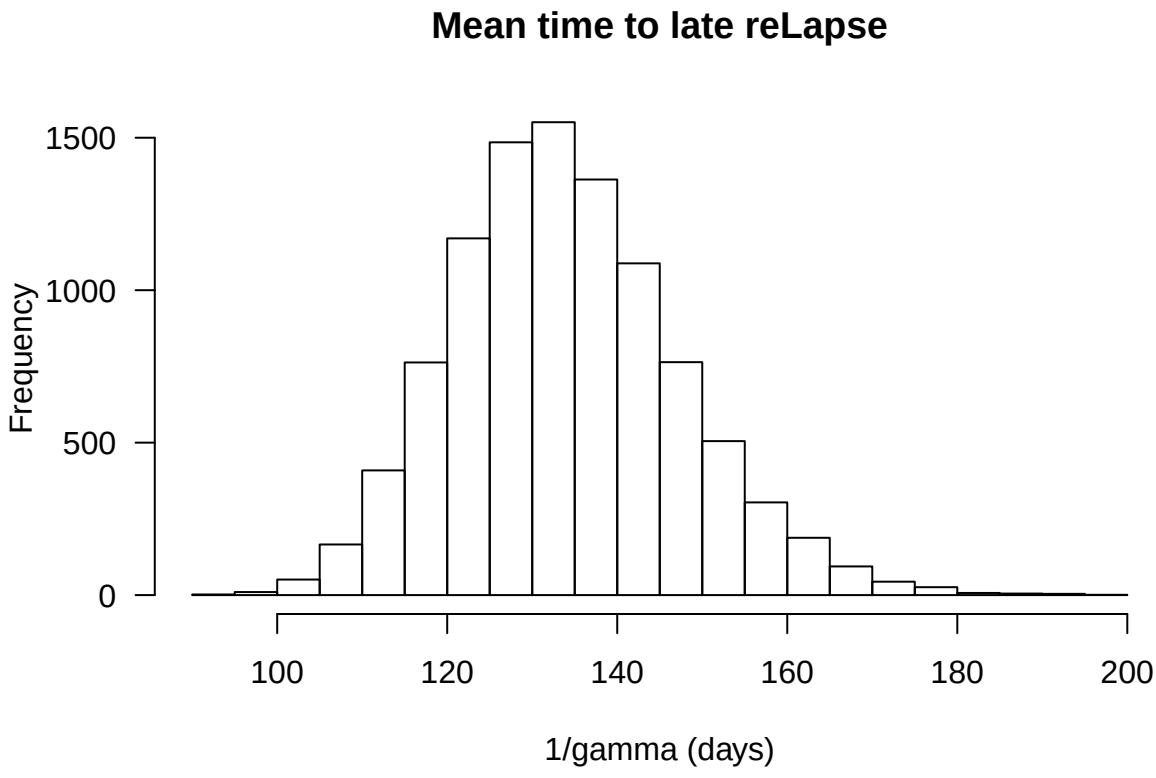
```
par(las=1)
traceplot(mod2, c('inv_lambda','Early_L','p','inv_gamma'))
```



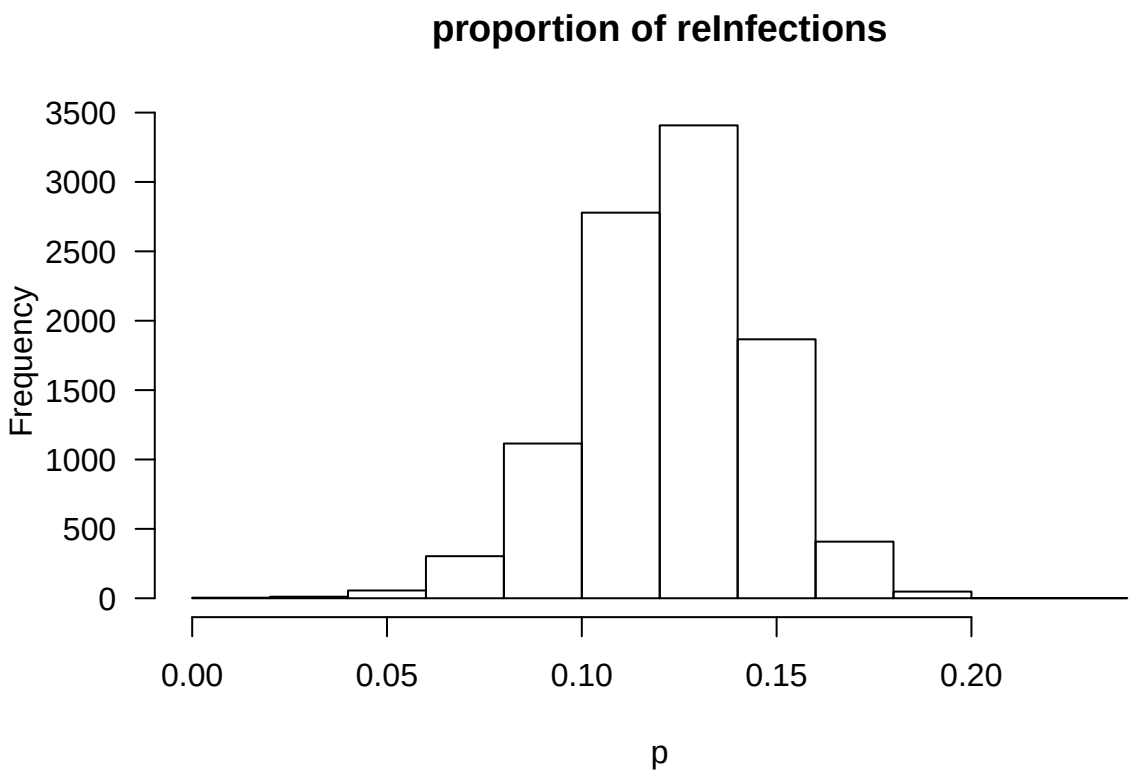

```
thetas = extract(mod2)
hist(thetas$inv_lambda, main='Mean time to reinfection', xlab='1/lambda (days)')
```



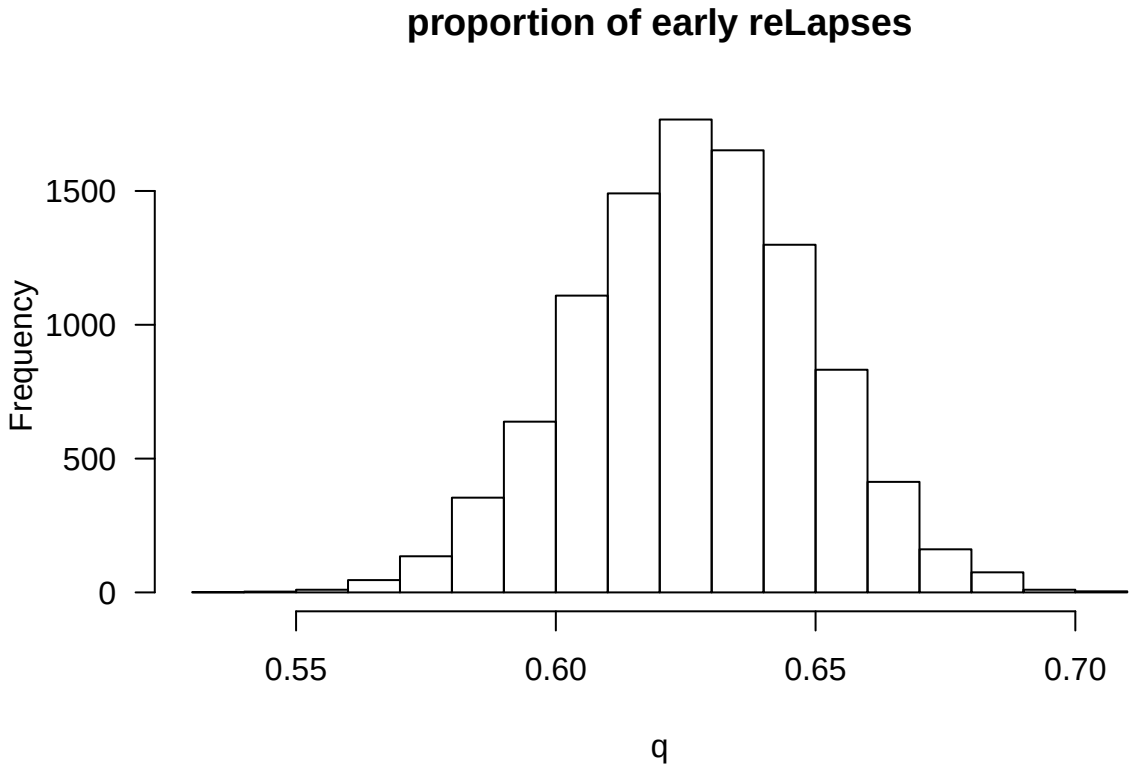

```r
hist(thetas$inv_gamma, main='Mean time to late reLapse', xlab='1/gamma (days)')
```




```r
hist(thetas$p, main = 'proportion of reInfections', xlab = 'p')
```

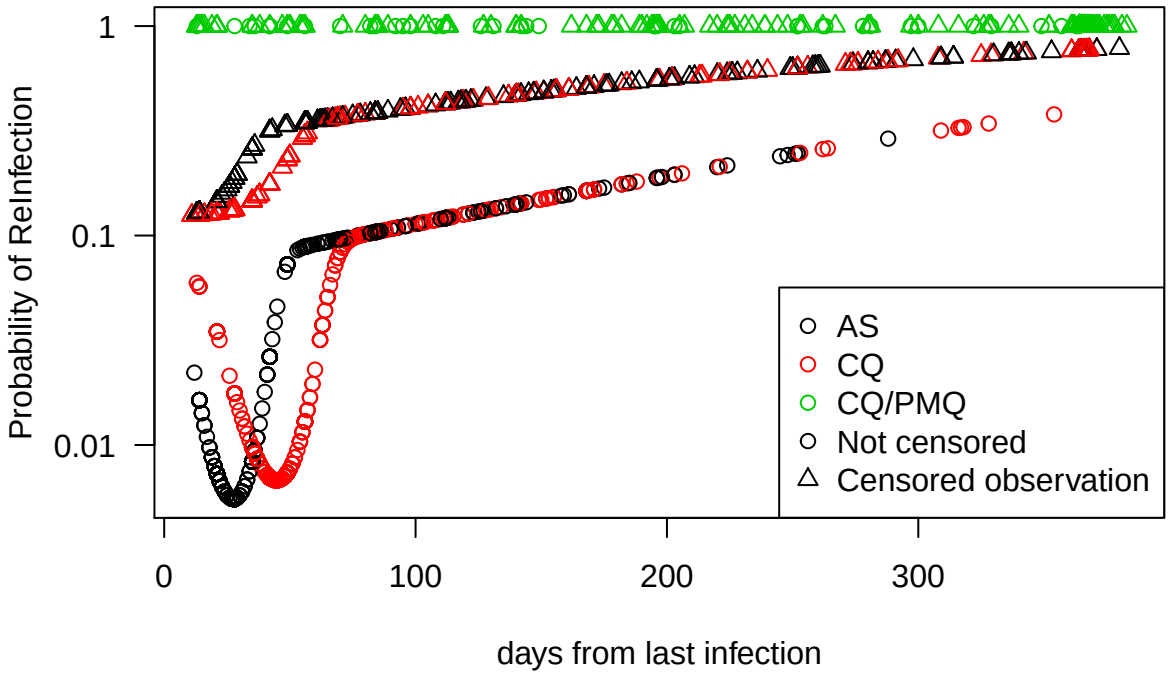



```
hist(thetas$Early_L, main = 'proportion of early reLapses', xlab='q')
```



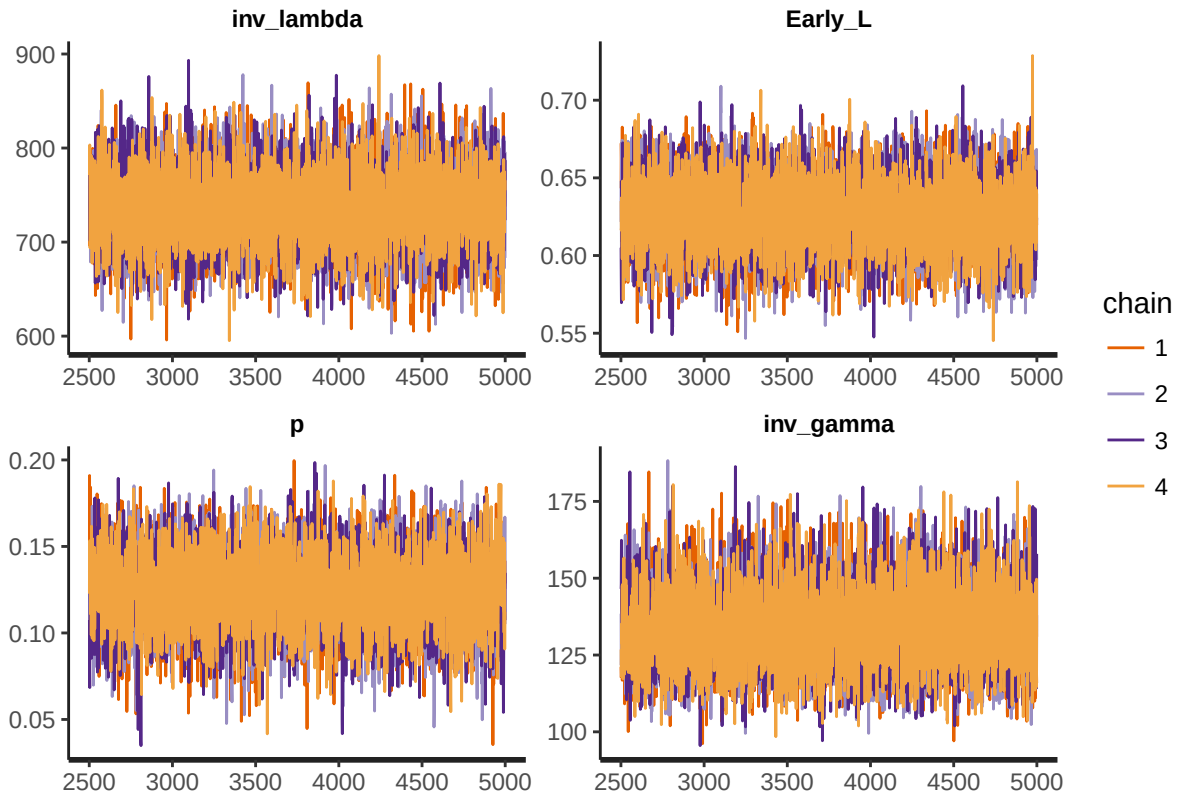

```
labels = extract(mod2, 'prob_labels')$prob_labels
mean_labels = apply(labels, c(2,3), mean)
plot(vhx_dat$Time_to_event, log10(mean_labels[,1]), col = numeric_drug+1, pch = as.numeric(vhx_dat$Righ
```

```
axis(2, at= -2:0, labels= 10^(-2:0))
legend('bottomright',legend = c('AS','CQ','CQ/PMQ', 'Not censored','Censored observation'), col=c(1:3,1
```

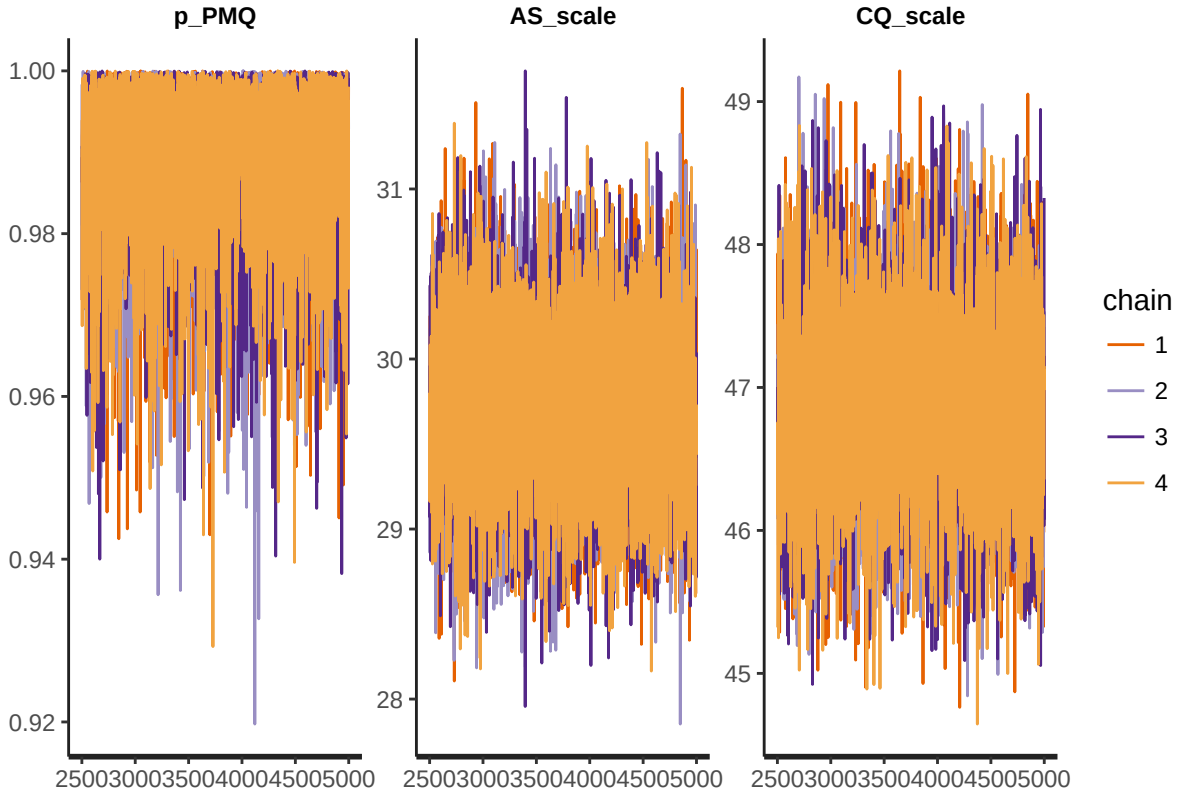


7.4 Model 3

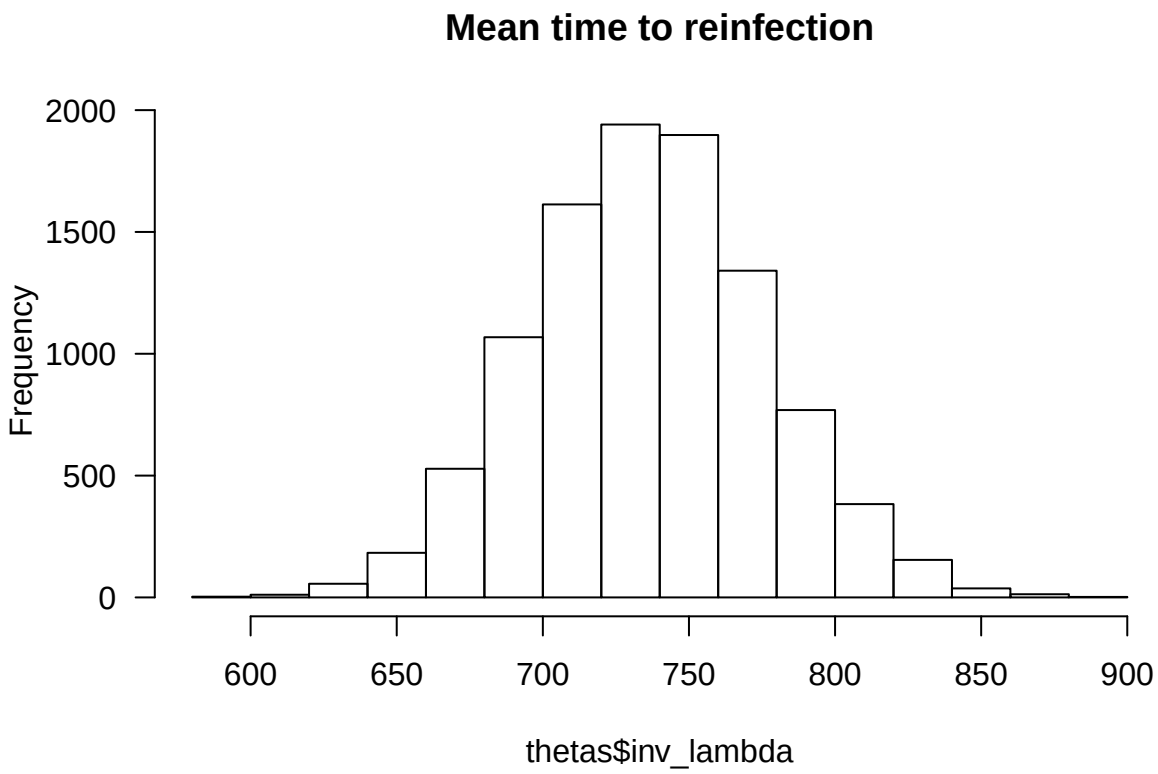
```
par(las=1)
traceplot(mod3, c('inv_lambda','Early_L','p','inv_gamma'))
```



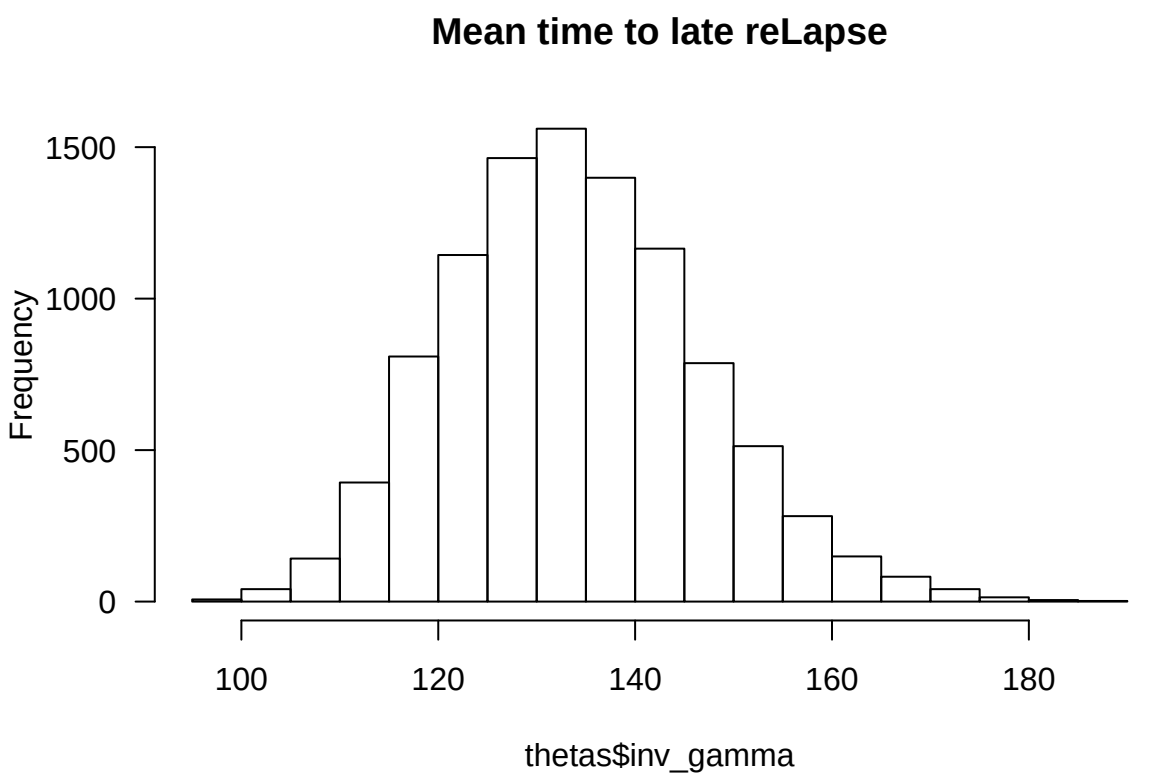

```
traceplot(mod3, c('p_PMQ','AS_scale','CQ_scale'))
```



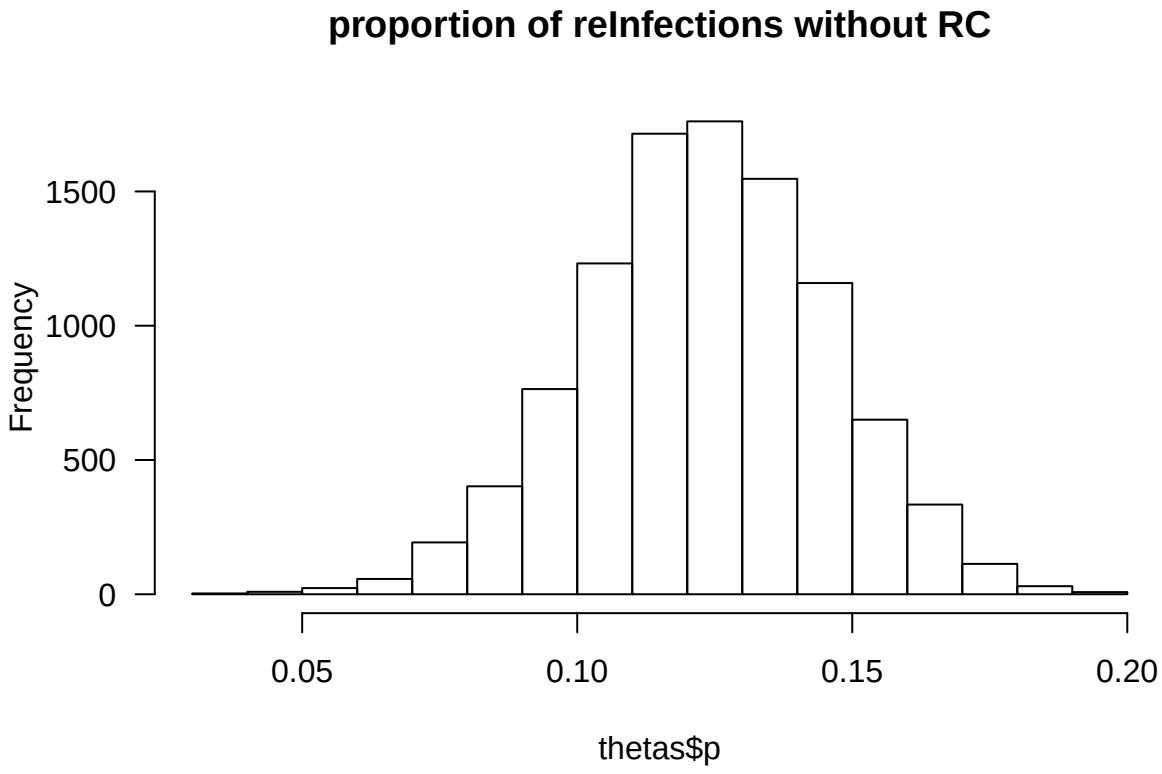

```
thetas = extract(mod3)
hist(thetas$inv_lambda, main='Mean time to reinfection')
```



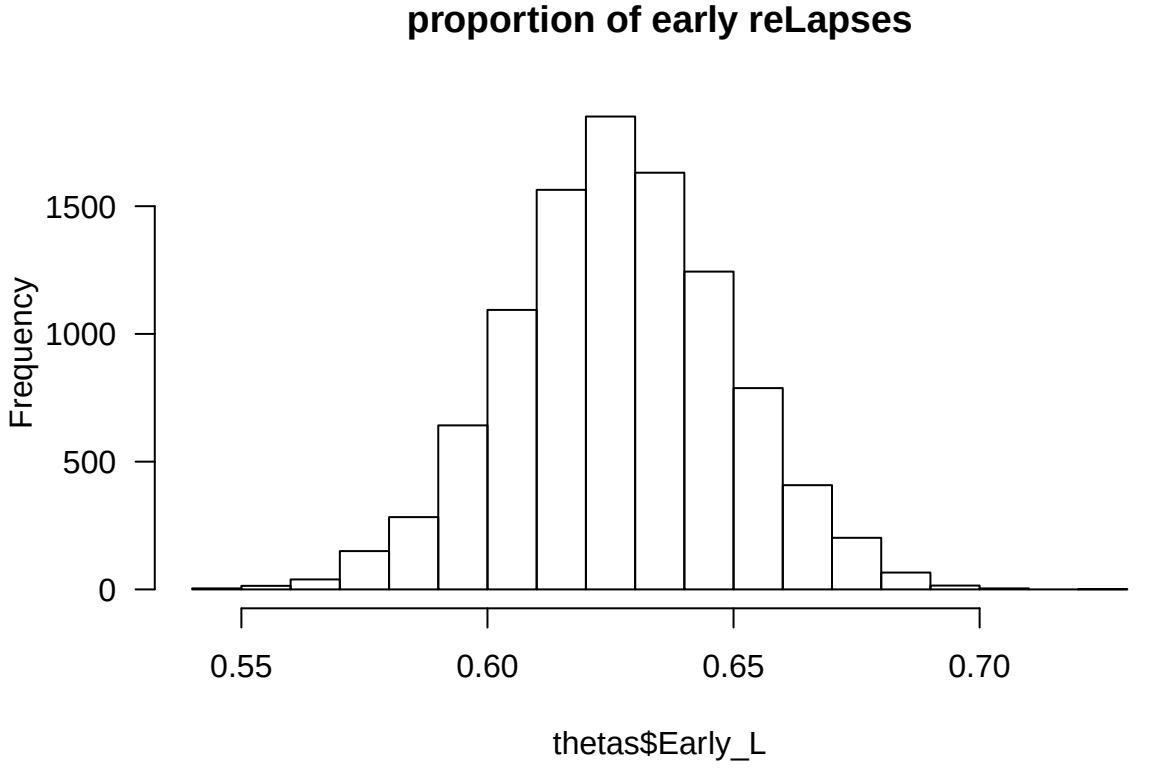

```
hist(thetas$inv_gamma, main='Mean time to late reLapse')
```



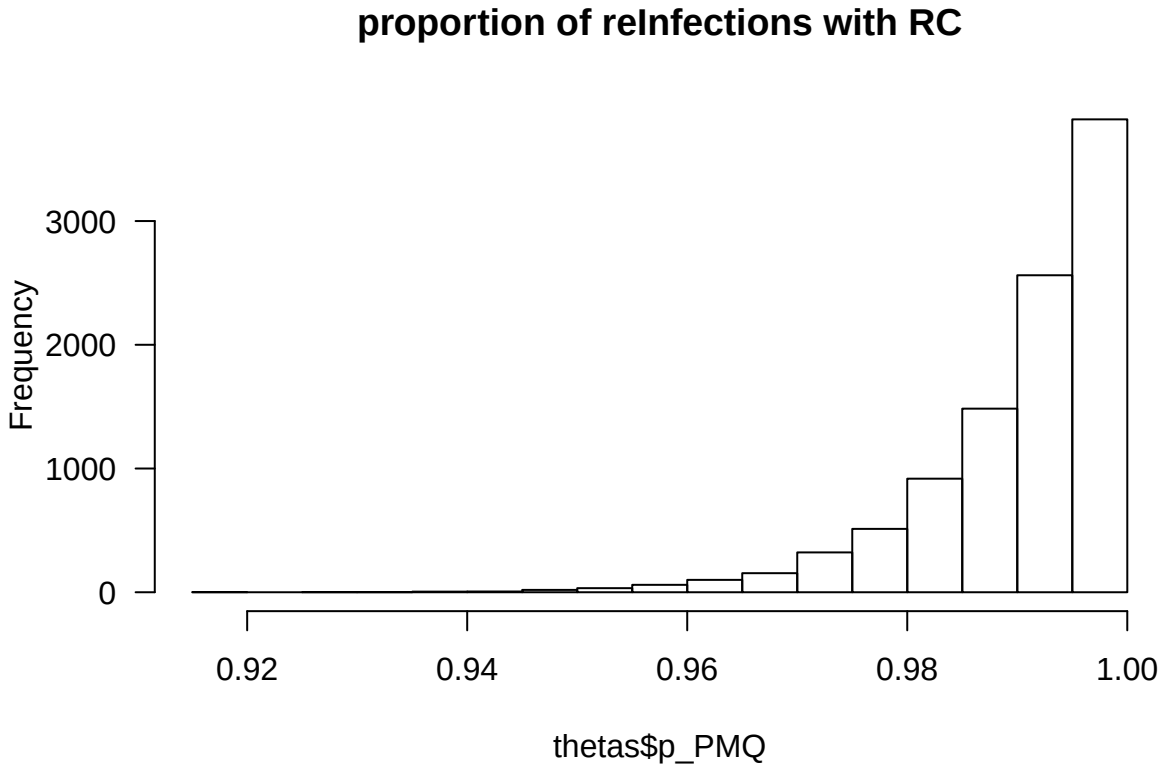

```
hist(thetas$p, main = 'proportion of reInfections without RC')
```



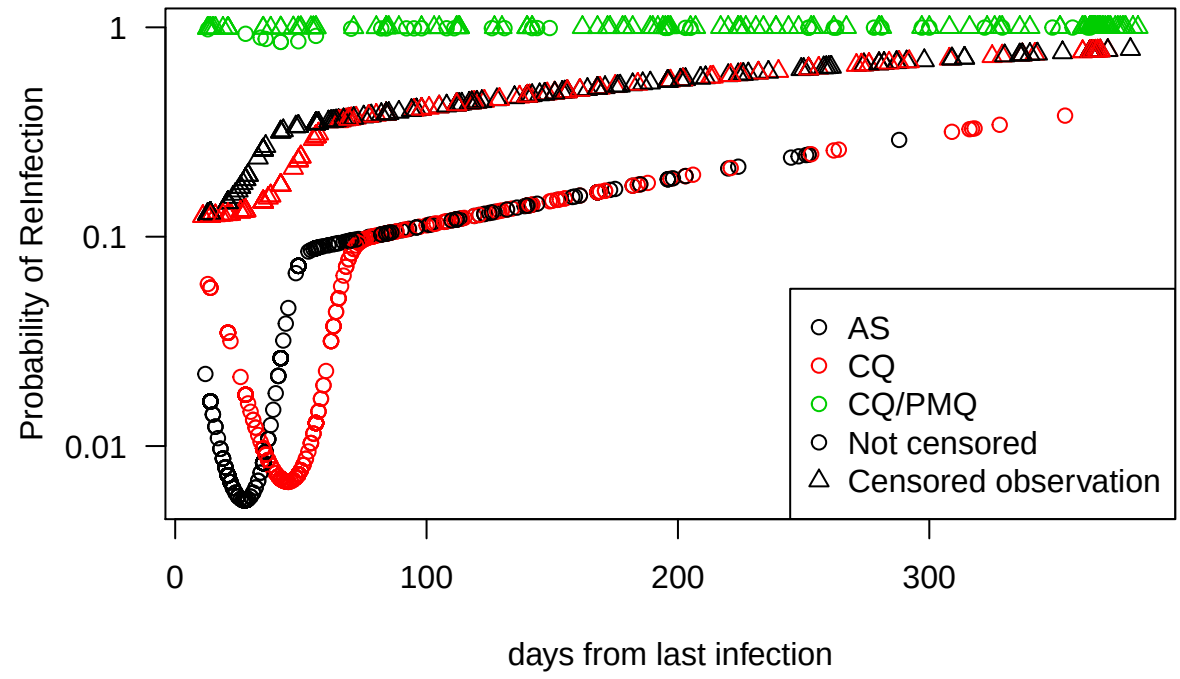

```
hist(thetas$Early_L, main = 'proportion of early reLapses')
```




```
hist(thetas$p_PMQ, main = 'proportion of reInfections with RC')
```

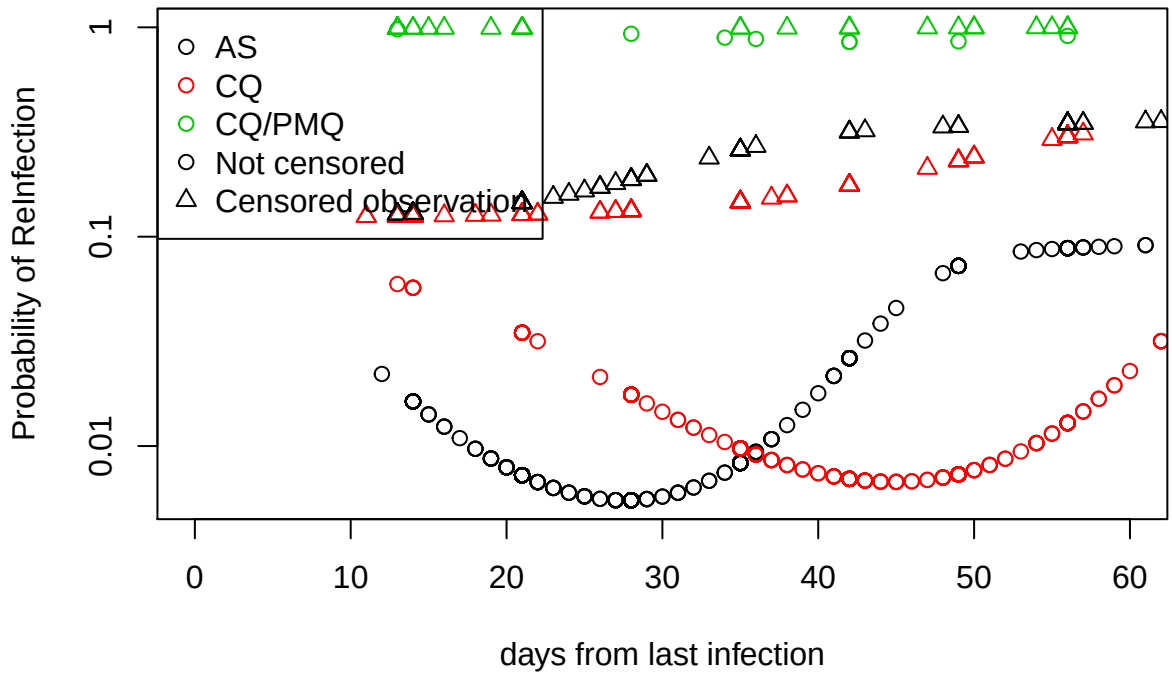



```
labels = extract(mod3, 'prob_labels')$prob_labels
mean_labels = apply(labels, c(2,3), mean)
plot(vhx_dat$Time_to_event, log10(mean_labels[,1]), col = numeric_drug+1, pch = as.numeric(vhx_dat$Righ
axis(2, at= -2:0, labels= 10^(-2:0))
legend('bottomright',legend = c('AS','CQ','CQ/PMQ', 'Not censored','Censored observation'), col=c(1:3,1
```



Zoomed in version

```
plot(vhx_dat$Time_to_event, log10(mean_labels[,1]),
     col = numeric_drug+1, xlim=c(0,60),
     pch = as.numeric(vhx_dat$Right_Truncated)+1,
     ylab='Probability of ReInfection',
     xlab='days from last infection',yaxt='n')
axis(2, at= -2:0, labels= 10^(-2:0))
legend('topleft',legend = c('AS','CQ','CQ/PMQ', 'Not censored','Censored observation'),
       col=c(1:3,1,1),pch = c(rep(1,3),1:2))
```



8 Model Comparison

```
library(loo)
```

```
## This is loo version 1.1.0
```

```
log_lik0 <- extract_log_lik(mod0)
log_lik1 <- extract_log_lik(mod1)
log_lik2 <- extract_log_lik(mod2)
log_lik3 <- extract_log_lik(mod3)
waic1 = waic(log_lik1)
loo1 = loo(log_lik1)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-
## diagnostic') for details.
```

```
waic2 = waic(log_lik2)
loo2 = loo(log_lik2)
waic3 = waic(log_lik3)
loo3 = loo(log_lik3)
compare(waic1, waic2, waic3)
```

```
##       waic    se_waic elpd_waic se_elpd_waic p_waic  se_p_waic
## waic2 13789.8   182.1 -6894.9      91.0          7.0     0.2
## waic3 13791.0   182.0 -6895.5      91.0          7.0     0.2
## waic1 13976.1   188.9 -6988.1      94.5         12.0     0.6
```

```
compare(loo1, loo2, loo3)
```

```
##      looic   se_looic elpd_loo se_elpd_loo p_loo   se_p_loo
## loo2 13789.8   182.1  -6894.9     91.0         7.0     0.2
## loo3 13791.0   182.0  -6895.5     91.0         7.0     0.2
## loo1 13977.7   189.0  -6988.8     94.5        12.8     0.7
```