Solutions for assignment 8

Guy Roberts

2023-03-23

Solution to 1.1

```
data {
  int<lower=0> N;
  int<lower=0> h[N];
}

parameters {
  real<lower=0> lambda;
}

model {
  // Prior
  lambda ~ gamma(2, 2);

  // Likelihood
  for (i in 1:N) {
    h[i] ~ poisson(lambda);
  }
}

generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = poisson_lpmf(h[i] | lambda);
}
```

```
data {
  int<lower=0> N;
  int<lower=0> h[N];
}

parameters {
  real<lower=0> lambda;
}

model {
  // Prior
  lambda ~ gamma(2, 2);

```

```
  // Likelihood
  for (i in 1:N) {
    h[i] ~ binomial(100, lambda/100);
  }
}

generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = binomial_lpmf(h[i] | 100, lambda/100);
}
```

```
data {
  int<lower=0> N;
  vector[N] h;
}

parameters {
  real<lower=0> lambda;
}

model {
  // Prior
  lambda ~ gamma(2, 2);

  // Likelihood
  for (i in 1:N) {
    h[i] ~ normal(lambda, (lambda*(100-lambda))/100);
  }
}

generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = normal_lpdf(h[i] | lambda, (lambda*(100-lambda))/100);
}
```

Solution to 1.2

```r
library(rstan)
library(ggplot2)
our_data = list(N = 20,
                h = c(1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0))
setwd(
  'C:/Users/guyro/OneDrive/Y3/bayesian-modelling/assignments/assignment8'
)

# Load the Stan models into R
model1 = stan_model('model1.stan')
model2 = stan_model('model2.stan')
model3 = stan_model('model3.stan')
```

```r
# Sample from the posterior (fit the models)
fit1 = sampling(model1, our_data, iter = 10000)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000521 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 5.21 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.083 seconds (Warm-up)
## Chain 1:                0.095 seconds (Sampling)
## Chain 1:                0.178 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.083 seconds (Warm-up)
## Chain 2:                0.082 seconds (Sampling)
## Chain 2:                0.165 seconds (Total)
## Chain 2:
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 5e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.081 seconds (Warm-up)
## Chain 3:                0.089 seconds (Sampling)
## Chain 3:                0.17 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 4e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.087 seconds (Warm-up)
## Chain 4:                0.081 seconds (Sampling)
## Chain 4:                0.168 seconds (Total)
## Chain 4:
```

```r
fit2 = sampling(model2, our_data, iter = 10000)
```

```
##
```

```
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000258 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.58 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.1 seconds (Warm-up)
## Chain 1:                0.103 seconds (Sampling)
## Chain 1:                0.203 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.099 seconds (Warm-up)
## Chain 2:                0.094 seconds (Sampling)
## Chain 2:                0.193 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 6e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.06 seconds.
```

```
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.117 seconds (Warm-up)
## Chain 3:                0.125 seconds (Sampling)
## Chain 3:                0.242 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 7e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.101 seconds (Warm-up)
## Chain 4:                0.107 seconds (Sampling)
## Chain 4:                0.208 seconds (Total)
## Chain 4:
```

```
fit3 = sampling(model3, our_data, iter = 10000)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.4e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.24 seconds.
## Chain 1: Adjust your expectations accordingly!
```

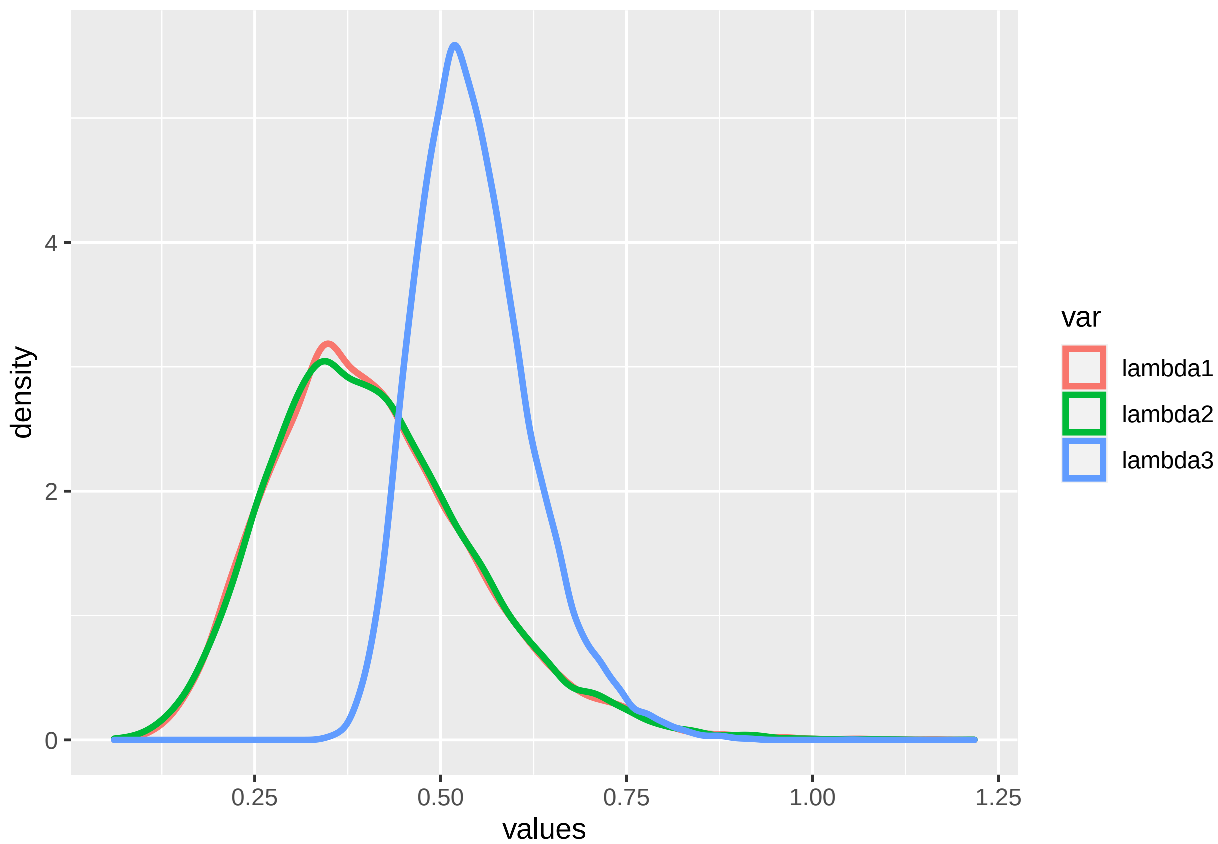
```
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.094 seconds (Warm-up)
## Chain 1:                0.101 seconds (Sampling)
## Chain 1:                0.195 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.1e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.095 seconds (Warm-up)
## Chain 2:                0.105 seconds (Sampling)
## Chain 2:                0.2 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 6e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.06 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 3: Iteration: 1000 / 10000 [ 10%]  (Warmup)
```

```
## Chain 3: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 3: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 3: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 3: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 3: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 3: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 3: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 3: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 3: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 3: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.095 seconds (Warm-up)
## Chain 3:                0.138 seconds (Sampling)
## Chain 3:                0.233 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 7e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 4: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 4: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 4: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 4: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 4: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 4: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 4: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 4: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 4: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 4: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 4: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.096 seconds (Warm-up)
## Chain 4:                0.125 seconds (Sampling)
## Chain 4:                0.221 seconds (Total)
## Chain 4:
```

```r
# Extract the lambdas
lambda1 = extract(fit1)$lambda
lambda2 = extract(fit2)$lambda
lambda3 = extract(fit3)$lambda

# Plot the posterior samples
dat = data.frame(var = factor(rep(c(
  'lambda1', 'lambda2', 'lambda3'
), each=20000)),
values = c(lambda1, lambda2, lambda3))

ggplot(dat, aes(x = values, color=var)) + geom_density(size=1.15)
```



Solution to 1.3

```
library(loo)
# Calculate pseudo-BMA weights
loo1 = loo(fit1)
loo2 = loo(fit2)
loo3 = loo(fit3)
```

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

```
# Print some of the weights
loo2
```

```
##
## Computed from 20000 by 20 log-likelihood matrix
##
##          Estimate  SE
## elpd_loo    -16.0 3.1
## p_loo         0.8 0.3
## looic        32.0 6.3
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
```

```
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```r
loo3
```

```
##
## Computed from 20000 by 20 log-likelihood matrix
##
##          Estimate  SE
## elpd_loo    -20.5 4.7
## p_loo         2.3 1.9
## looic        41.0 9.4
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     19    95.0%   6166
##  (0.5, 0.7]   (ok)        0     0.0%   <NA>
##    (0.7, 1]   (bad)       1     5.0%   133
##    (1, Inf)   (very bad)  0     0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

```r
loo_model_weights(list(loo1, loo2, loo3),
                  method = "pseudobma",
                  BB = FALSE)
```

```
## Method: pseudo-BMA
## ------
##        weight
## model1 0.505
## model2 0.490
## model3 0.005
```

- This link provides some insight on the output from the LOO fits.

NOTE: We get a warning about some Pareto k diagnostic values being slightly high - One value is in the range $(0.5, 0.7]$ (ok) : which may indicate that the model is badly mis-specified (see link).

Considering the sample densities of the posterior (shown above), these results do match our expectations. The graph shows that models 1 and 2 are very similar, with mode ~ 0.4, whereas the mode for model 3 ~ 0.5 and this model has a lower spread than the others, and hence a higher peak. The results also suggest that model 1 is marginally more suitable (given the data) than model 2 - however, the dataset used is small, so perhaps more in-depth analysis is required in order to make an appropriate decision.

Solution to 2.1

Here are the models:

$$
\begin{aligned}
M_1 : &\quad Y|\alpha \sim \text{Exp}(\alpha) \\
M_2 : &\quad Y|\nu \sim \chi^2(\nu) \\
M_3 : &\quad \sqrt{Y}|\beta \sim \text{Maxwell-Boltzmann}(\beta)
\end{aligned}
$$

Now if we consider the PDFs of each distribution:

$$\pi_{M_1}(y) \propto \exp(-y) \tag{1}$$
$$\pi_{M_2}(y) \propto y^{\frac{\nu}{2}-1} \exp(-\frac{y}{2}) \tag{2}$$
$$\pi_{M_3}(y) \propto y \exp(-y) \tag{3}$$

Then we can see that the similarity between these distributions is that they are can all be formed from Gamma distributions: Gamma(1, 1), Gamma($\frac{\nu}{2}$, $\frac{1}{2}$) and Gamma(2, 1) respectively.

Solution to 2.2

As mentioned, the priors could be Gamma(1, 1), Gamma($\frac{\nu}{2}$, $\frac{1}{2}$) and Gamma(2, 1).

```
data {
  int<lower=0> N; // num obs
}

parameters {
  real<lower=0> alpha;          // rate param for exp distribution
  real<lower=0> nu;             // df param for chisq distribution
  real<lower=0> beta;           // scale parameter for MB distribution
  real<lower=0,upper=1> p1;     // exp distribution (mixing prop.)
  real<lower=0,upper=1-p1> p2;  // chisq distribution (mixing prop.)
}

model {
  // Priors
  alpha ~ gamma(1, 1);
  nu ~ gamma(0.5*nu, 0.5);
  beta ~ gamma(2, 1);
}

generated quantities {
  real<lower=0> y_sim[N];

  for (n in 1:N) {
    if (uniform_rng(0, 1) < p1) {
      // exponential
      y_sim[n] = exponential_rng(alpha);
    } else if (uniform_rng(0, 1) < p1 + p2) {
      // chi-squared
      y_sim[n] = chi_square_rng(nu);
    } else {
      // Maxwell-Boltzmann
      real z = normal_rng(0, 1);
      real y = sqrt(2 / (pi() * beta)) * exp(-z^2 / 2);
      y_sim[n] = y^2; // squared since we are given sqrt(Y) ~ MB
    }
  }
}
```

```r
model4 = stan_model('model4.stan')

num_samples = 1000
#samples = sampling(model4, data=list(N=num_samples), iter=10000)

#y_sample = extract(samples, par='y_sim')$y_sim
#summary(y_sample)
```

This code is extremely slow and seems to give the wrong output - I haven't had enough time to fully understand what is happening hence the incompleteness. So it is commented out.

We can additionally modify the priors so that they accommodate this new information. For example, for the exponential distribution in M_1 we can use a Gamma(2, 1) which will give the prior belief that Y is on average 2.