

# Assignment 1

*Jaehyun Song*

*7/29/2017*

## Problem 1: Bayes' Rule (Theory)

Rewrite Bayes' Rule for two events,  $A$  and  $B$ , that are independent. How do you interpret this result?

Answer

$p(A)$  is a probability that the event  $A$  will occur and  $p(A|B)$  is a probability of  $A$  when event  $B$  observed.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (1)$$

If  $A$  and  $B$  are independent,  $P(A, B)$  is  $P(A)P(B)$ . In this case,

$$\begin{aligned} P(A|B) &= \frac{P(B, A)(= P(B|A)P(A))}{P(B)} \\ &= \frac{P(B)P(A)}{P(B)} \\ &= P(A). \end{aligned} \quad (2)$$

Therefore, the probability of  $A$  given  $B$  is equivalent to the probability of  $A$ . In this case, Bayes' rule leads to uninteresting result.

## Problem 2: Bayes' Rule (Practice)

Jeff is hoping to play baseball with his friends tomorrow, but will only play if it does not rain. Let  $R$  be the event of rain tomorrow,  $S$  be the event of sunshine tomorrow, and  $F$  be the event of the weather forecast predicting rain. The weather forecaster is 80% accurate in her career, meaning that the probability of  $R$  when the forecaster predicts sun and the probability of  $S$  when the forecaster predicts rain are both 20%. The historical probability of rain on this particular day is 10%.

- (a) Using Bayes' Rule, calculate the probability that it rains, given that the weather forecast predicts rain.

Answer

The probability which we are interested in is  $P(R|F)$ .  $P(R|F)$  can be transformed as Equation 3.

$$\begin{aligned} P(R|F) &= \frac{P(F|R)P(R)}{P(F)} \\ &= \frac{P(F|R)P(R)}{P(F|R)P(R) + P(S|R)P(R)}. \end{aligned} \quad (3)$$

We already know  $P(R) = 0.1$ ,  $P(S) = 1 - P(R) = 0.9$ ,  $P(F|R) = 0.8$ , and  $P(S|R) = 1 - P(R|F) = 0.2$ . Thus,

$$\begin{aligned} P(R|F) &= \frac{0.8 \cdot 0.1}{0.8 \cdot 0.1 + 0.2 \cdot 0.9} \\ &= \frac{0.08}{0.08 + 0.18} = \frac{0.08}{0.26} \approx 0.308, \end{aligned} \quad (4)$$

According Equation 4, The probability of rain when the weather forecaster predicts rain is about 30.8%.

- (b) In this simple example what represents your prior belief and what represents your data?

Answer

- Data: A fact that the weather forecaster predicted rain,  $P(F)$ .
- Prior belief: The historical probability of rain on the day,  $P(R)$ .

- (c) Bonus: Confirm this result using simulation in R.

Answer

```
# Set the seed for replication.  
set.seed(1986-10-08)
```

```

# Aaaa...
Rain <- sample(c("Rain", "Sun"), 10000, prob = c(0.1, 0.9), replace = TRUE)
# Create a vector to store the forecaster's prediction.
Predicted <- rep(NA, length(Rain))

# Create the forecaster's prediction records
for (i in 1:length(Rain)) {
  # If rain...
  if (Rain[i] == "Rain") {
    # The forecaster predicts rain with probability of 0.8
    # The forecaster predicts sun with probability of 0.2
    Predicted[i] <- sample(c("Rain", "Sun"), 1, prob = c(0.8, 0.2))
  }else{ # If not...
    # The forecaster predicts sun with probability of 0.8
    # The forecaster predicts rain with probability of 0.2
    Predicted[i] <- sample(c("Rain", "Sun"), 1, prob = c(0.2, 0.8))
  }
}

Q2.tbl <- table(Rain, Predicted)

Q2.tbl

```

```

##          Predicted
## Rain   Rain  Sun
##   Rain  778 199
##   Sun  1763 7260

```

In the simulation, the weather forecaster predicted rain 2541 times (= 778 + 1763), and it actually rained 778 times. The probability of rain given that the forecaster predicts rain is approximately  $\frac{778}{2541} \approx 0.306$ , 30.6%.

## Problem 3: Experimenting with Priors

### Problem 3.1

Use R to create plots of different variations on the Beta distribution.

- (a) Plot at least 3 variations with different means and variances. Include all three distributions on the same graph. Distinguish the distributions using different colors and include a legend.

Answer

I show three different Beta distribution with (1) mean of 0.5 and variance of 0.083, (2) mean of 0.75 and variance of 0.05, and (3) mean of 0.35 and variance of 0.003.

Given mean and variance, the parameters of Beta distribution can be obtained as follow:

$$\begin{aligned}\alpha &= \left(\frac{1-\mu}{\sigma^2} - \frac{1}{\mu}\right)\mu^2 \\ \beta &= a\left(\frac{1}{\mu} - 1\right).\end{aligned}\tag{5}$$

```
# Create a function to calculate the parameters of Beta distribution
# given mean and variance.
CalcBetaParm <- function(mean, var) {
  alpha <- ((1 - mean) / var - 1 / mean) * mean ^ 2
  beta <- alpha * (1 / mean - 1)
  return(c(alpha, beta))
}

# Get the parameters
Param1 <- CalcBetaParm(0.5, 0.083) # Prior1
Param2 <- CalcBetaParm(0.66, 0.050) # Prior2
Param3 <- CalcBetaParm(0.35, 0.003) # Prior3

# Prior1: mean = 0.5, variance = 0.083
Prior1 <- dbeta(seq(0, 1, 0.01), Param1[1], Param1[2])
# Prior2: mean = 0.66, variance = 0.050
Prior2 <- dbeta(seq(0, 1, 0.01), Param2[1], Param2[2])
# Prior3: mean = 0.35, variance = 0.002
Prior3 <- dbeta(seq(0, 1, 0.01), Param3[1], Param3[2])

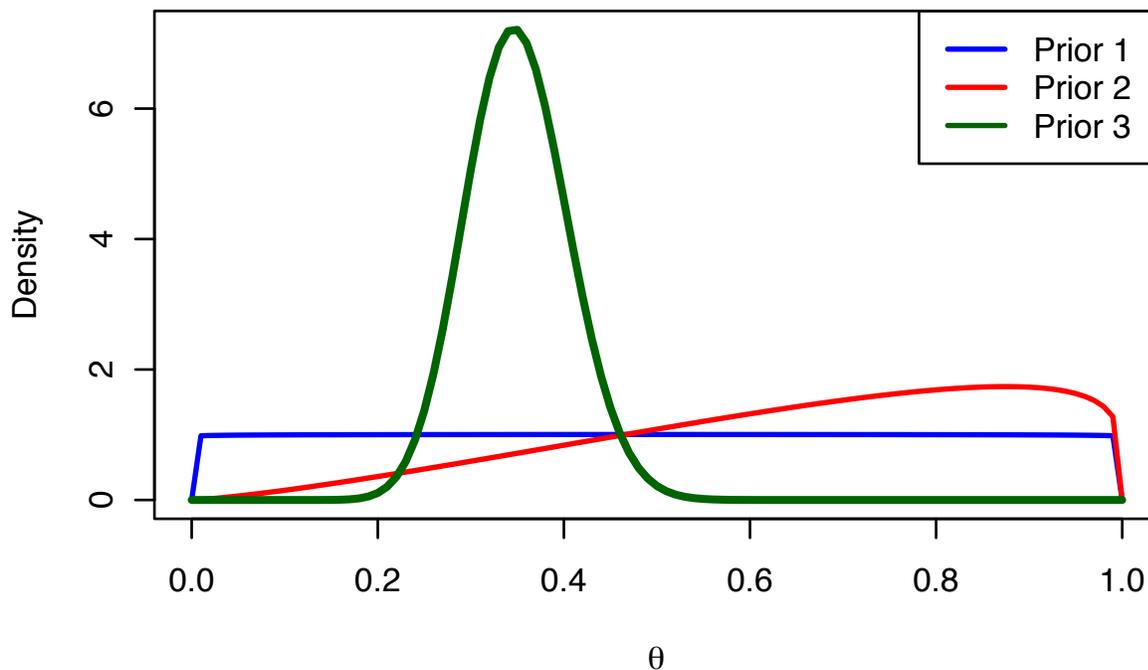
# Plot the priors
plot(x = seq(0, 1, 0.01), y = Prior1,
      type = "l", col = "blue", lwd = 2,
```

```

ylim = c(0, max(c(Prior1, Prior2, Prior3))),
xlab = expression(theta), ylab = "Density",
main = "Three prior distributions")
lines(x = seq(0, 1, 0.01), y = Prior2, col = "red", lwd = 2)
lines(x = seq(0, 1, 0.01), y = Prior3, col = "darkgreen", lwd = 3)
legend("topright", col = c("blue", "red", "darkgreen"), lwd = 2,
      legend = paste("Prior", 1:3))

```

Three prior distributions



- (b) Discuss what type of prior each form would represent (e.g. is it a strong prior? A weak prior? An uninformative prior?).
- (c) For 1 of the distributions plotted, identify and explain a scenario in which that prior might be appropriate.

Answer to (b) and (c)

For example, suppose that I want to approval rate for LDP<sup>1</sup> in July 2017. If I lack the knowledge about political science, contemporary Japanese politics, what party is in government, or how many parties there are in Japan, I need to treat all possibility equally. In this case, uninformative prior, uniform distribution ( $Beta(1, 1)$ ), is plausible.

If I read an introductory textbook on Japanese politics, and know that Japanese party system has been called *one and one-half* system for a long time. LDP had governed Japan from 1955 to 1993, it is plausible to consider LDP was the strongest party in Japan. Thus, I set the mean of prior distribution  $0.66 (= \frac{2}{3})$ . However, it is 2017 now. I cannot say LDP is still

<sup>1</sup>Liberal Demoratic Party (*Jiminto*). The biggest and governance party in Japan.

majority with confidence. Thus, relatively diffused distribution is proper, for example, 0.05. In this case, I adopt a weak prior,  $Beta(2.30, 1.19)$

If I am a specialist in Japanese politics, approval rates for LDP in June is available. The table below is actual approval rates for LDP in June conducted by 11 different research institution<sup>2</sup>. It is a kind of strong prior distribution.

Institution	Support for LDP
Asahi	0.34
Yomiuri	0.40
Mainichi	0.27
Nikkei	0.40
Kyodo	0.34
Jiji	0.25
NHK	0.36
Nihon TV	0.39
JNN	0.34
FNN	0.36
ANN	0.40
<b>Mean</b>	0.350
<b>Variance</b>	0.003

The mean and variance of approval rates is approximately 0.35 and 0.003, respectively. A Beta distribution which has mean of 0.35 and variance of 0.003 is  $Beta(26.19, 48.64)$ .

## Problem 3.2

Repeat the exercise above, but this time use the Gamma distribution.

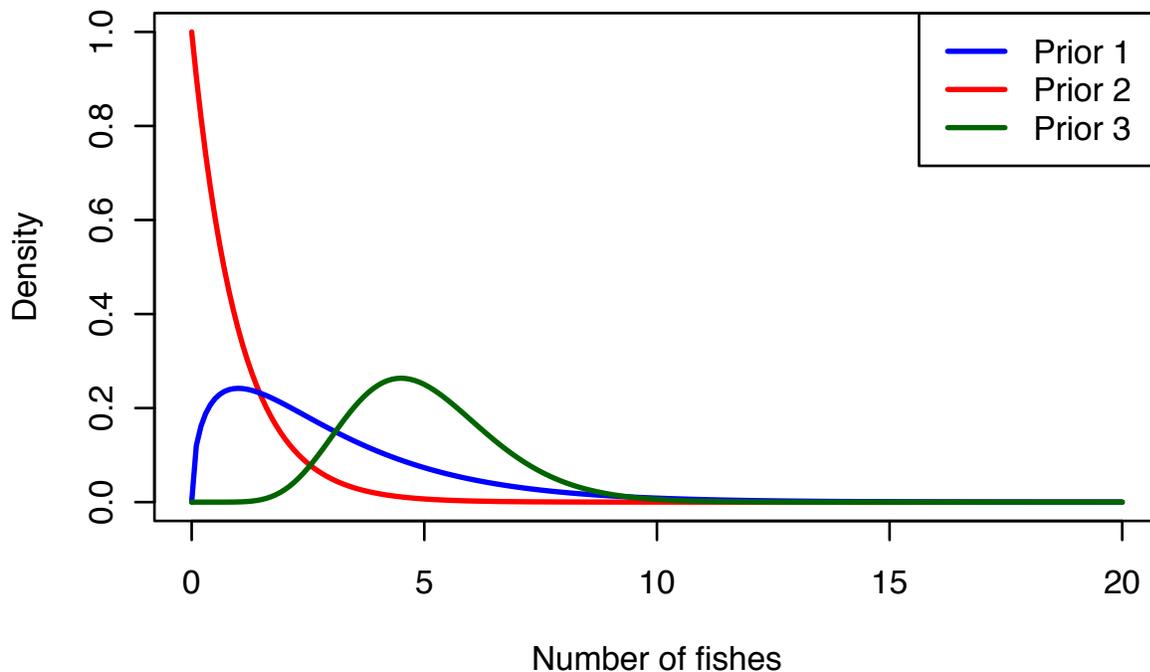
```
#Prior1: mean = 1, variance = 1
G.Prior1 <- dgamma(x = seq(0, 20, 0.1), shape = 1, scale = 1)
#Prior2: mean = 3, variance = 6
G.Prior2 <- dgamma(x = seq(0, 20, 0.1), shape = 1.5, scale = 2)
#Prior3: mean = 5, variance = 2.5
G.Prior3 <- dgamma(x = seq(0, 20, 0.1), shape = 10, scale = 0.5)

#Plot the three distribution
plot(x = seq(0, 20, 0.1), y = G.Prior1, type = "l", lwd = 2, col = "red",
      xlab = "Number of fishes", ylab = "Density",
      main = "Three different Gamma prior distributions")
lines(x = seq(0, 20, 0.1), y = G.Prior2, lwd = 2, col = "blue")
lines(x = seq(0, 20, 0.1), y = G.Prior3, lwd = 2, col = "darkgreen")
```

<sup>2</sup><http://www.realpolitics.jp/research/>

```
legend("topright", col = c("blue", "red", "darkgreen"), lwd = 2,
      legend = paste("Prior", 1:3))
```

### Three different Gamma prior distributions



These prior distributions are all weak prior distribution and the second prior ( $\mathcal{Gamma}(2, 2)$ ) is relatively the weakest prior because its variance is the largest among the prior.

Suppose that three of my friends (Sohei, Keisuke, Yuki) went to fishing and I wait them. If they caught some fishes, I need to prepare for cooking. If they did not catch any fishes, I need to go to buy fishes. Assume how many fishes they will catch follows Poisson distribution. Gamma distribution is well known as the prior distribution of Poisson distribution.

Yuki has never experienced fishing and is very afraid of natural baits. I do not expect he will catch any fishes, thus, I assign prior  $\mathcal{Gamma}(1, 1)$  to Yuki's posterior (red line in the figure).

Keisuke has some experience, but not familiar with fishing. However, he caught ten fishes in the last month, I believe he has a talent for fishing. I assign prior  $\mathcal{Gamma}(1.5, 2)$  to his posterior. It has pretty diffused variance, it means he can catch many fishes in some cases.

Sohei is called *King of Fishing*. He caught a big whale last year. I am pretty sure he caught some fishes. He is worthy of  $\mathcal{Gamma}(10, 0.5)$ .

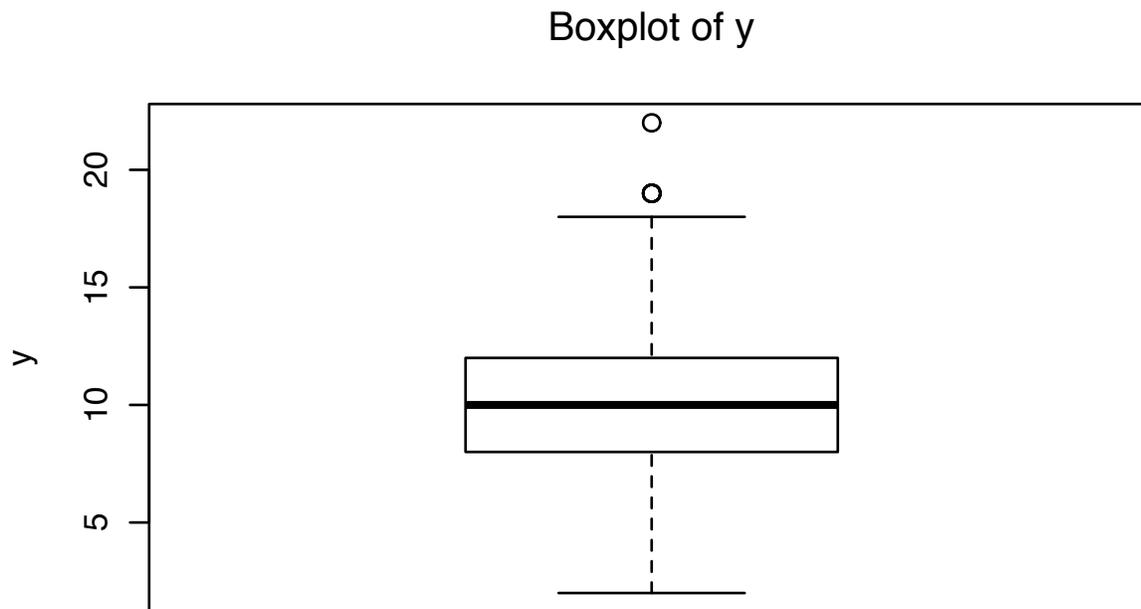
## Problem 4: The Poisson/Gamma Conjugacy

Use  $y$  from the `week1.csv` data file. Let  $y \sim \text{Poisson}(\lambda)$  and estimate  $\lambda$ .

- (a) First, estimate  $\lambda$  by coding and maximizing the Poisson likelihood. Include the point estimate and confidence intervals.

```
# Import the dataset
A1.df <- read.csv("Week1.csv")

# Check the distribution of variable, y.
boxplot(A1.df$y, ylab = "y", main = "Boxplot of y")
```



The maximum value of  $y$  is approximately 25,  $\lambda$  is expected smaller than 30. Thus, I calculate log likelihood in range of  $\lambda \in [0, 30]$  as follow:

```
# Create a vector contains value of lambda which I will calculate
# log likelihood
lambda.vec <- seq(0, 30, 0.01)
# Create a vector to store log likelihood
LogLik.vec <- rep(NA, length(lambda.vec))

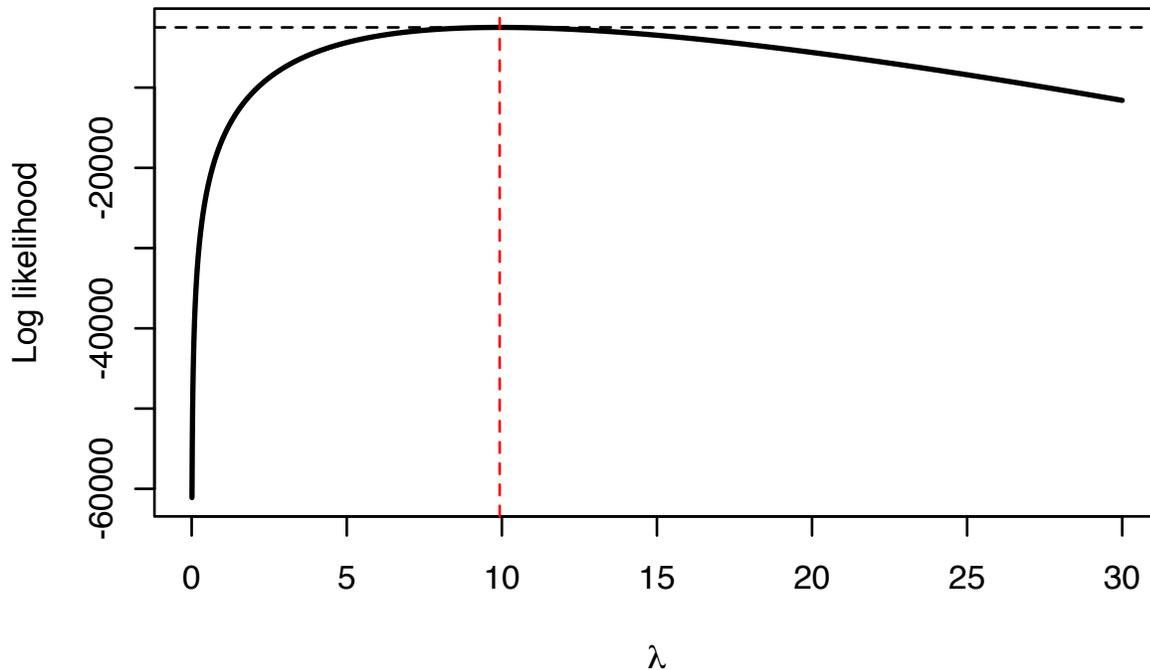
for (i in 1:length(lambda.vec)) {
  # Calculate and store log likelihood
  temp.LogLik <- sum(dpois(A1.df$y, lambda.vec[i], log = TRUE))
  LogLik.vec[i] <- temp.LogLik
}

# Plot the log likelihood
plot(x = lambda.vec, y = LogLik.vec, type = "l", lwd = 2,
```

```

xlab = expression(lambda), ylab = "Log likelihood")
abline(h = max(LogLik.vec), lty = 2)
abline(v = lambda.vec[which.max(LogLik.vec)], lty = 2, col = "red")

```



The value of  $\lambda$  which maximize log likelihood is 9.93. If  $\lambda = 9.93$ ,  $-\log$  likelihood is -2499.262. An approximate 95% interval of  $\lambda$  is,

$$95\% \text{ C.I} = \hat{\lambda} \pm 1.96 \sqrt{\frac{\hat{\lambda}}{n}}, \quad (6)$$

where  $n$  is sample size and  $\hat{\lambda}$  is estimated  $\lambda$ . Thus, 95% confidence intervals can be calculated as follow:

```

Max.LL <- lambda.vec[which.max(LogLik.vec)]

CI <- Max.LL + qnorm(c(0.025, 0.075)) * sqrt(Max.LL / nrow(A1.df))

cat("95% Confidence Intervals: [", CI[1], ", ", CI[2], "]", sep = "")

```

```
## 95% Confidence Intervals: [9.734691, 9.786552]
```

The 95% confidence intervals are [9.735, 9.787].

- (b) Using the conjugate Gamma prior, set the hyperparameters to represent a weak belief that  $\lambda \approx 20$ . This means the mean of the prior should be about 20, but there should be some variance around that mean. Fit the model to the data, describing the posterior distribution  $\lambda|y$  (point estimate and credible intervals).

Answer

First, specify the parameters of Gamma distribution. The mean of Gamma distribution is  $\frac{\alpha}{\beta}$  and variance is  $\frac{\alpha}{\beta^2}$ . To make its mean 20,  $20\alpha = \beta$ . If  $\beta = 1$ ,  $\alpha = 20$  and its variance is also 20.

Because Gamma distribution is a conjugate prior of Poisson distribution, posterior distribution given data  $y$  is

$$\lambda|y \sim \mathcal{Gamma}(\sum y_i + \alpha, n_y + \beta), \quad (7)$$

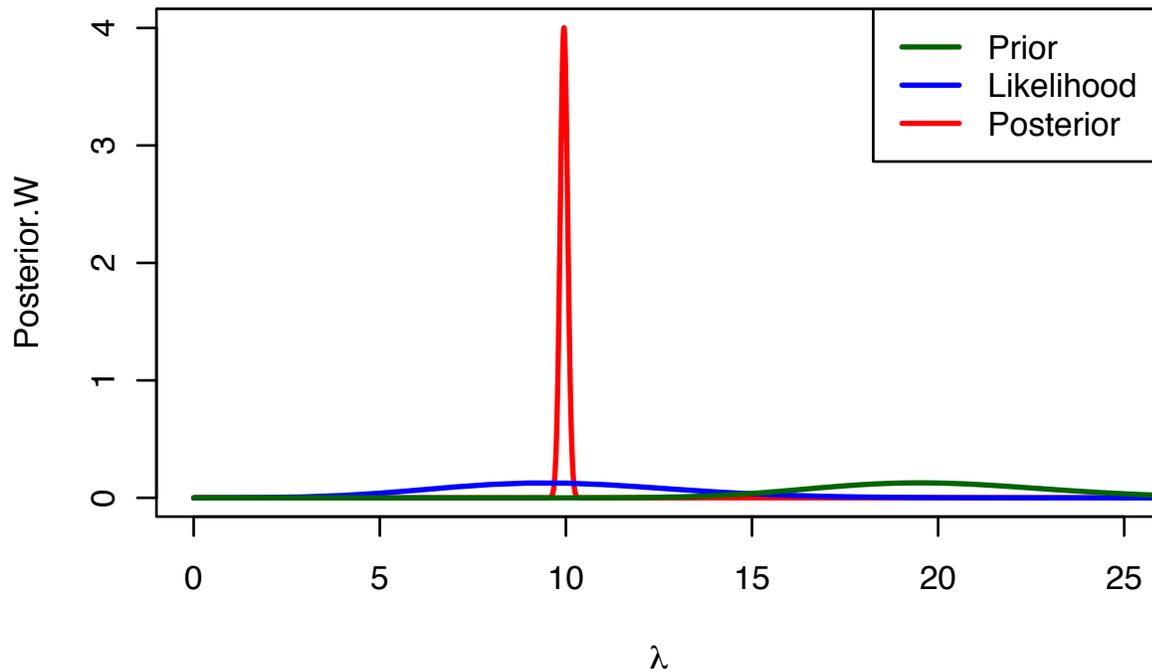
where  $\alpha$  and  $\beta$  are shape and rate parameters of Gamma distribution, respectively. This posterior distribution can be calculated easily using R.

```
# Set the parameters of prior and posterior distribution
prior.alpha <- 40
prior.beta  <- 2
post.alpha  <- sum(A1.df$y) + prior.alpha
post.beta   <- nrow(A1.df) + prior.beta

# Compute likelihood with parameter of lambda which I got from MLE.
Likelihood <- dpois(seq(0, 30, 1), lambda.vec[which.max(LogLik.vec)])
# Compute the density of prior distribution
Prior.W    <- dgamma(seq(0, 30, 0.01), prior.alpha, prior.beta)
# Compute the density of posterior distribution
Posterior.W <- dgamma(seq(0, 30, 0.01), post.alpha, post.beta)

# Plot the likelihood, prior and posterior distributions
plot(x = seq(0, 30, 0.01), y = Posterior.W, type = "l",
     lwd = 2, col = "red", xlim = c(0, 25),
     xlab = expression(lambda), main = "Posterior with a weak prior")
lines(x = seq(0, 30, 1), y = Likelihood,
      lwd = 2, col = "blue")
lines(x = seq(0, 30, 0.01), y = Prior.W,
      lwd = 2, col = "darkgreen")
legend("topright", lwd = 2, col = c("darkgreen", "blue", "red"),
      legend = c("Prior", "Likelihood", "Posterior"))
```

## Posterior with a weak prior



The EAP, MAP and 95% credible intervals can be calculated using `qgamma()` function.

```
# Compute and show EAP and MAP estimates.
```

```
Q4.EAP.W <- post.alpha / post.beta
```

```
Q4.MAP.W <- (post.alpha - 1) / post.beta
```

```
cat("EAP:", Q4.EAP.W)
```

```
## EAP: 9.949102
```

```
cat("MAP:", Q4.MAP.W)
```

```
## MAP: 9.948104
```

```
# Compute and show 95% credible intervals of the posterior distribution
```

```
Q4.CI.W <- qgamma(c(0.025, 0.975), post.alpha, post.beta)
```

```
cat("95% Credible Intervals: [", Q4.CI.W[1], ", ", Q4.CI.W[2], "]", sep = "")
```

```
## 95% Credible Intervals: [9.754747, 10.14535]
```

- EAP: 9.949
- MAP: 9.948
- 95% credible intervals: [9.755, 10.145]

HPD intervals can be calculated as:

```

Calc.HPDI <- function(x, y, hpdi = 0.95) {
  # Thanks, Hadely!
  library(dplyr)

  # Create a temporal data frame.
  temp.df <- data.frame(x, y)

  # Compute and show 95% HPDI
  temp.df <- temp.df %>% arrange(desc(y)) %>%
    mutate(cumy = cumsum(y),
           in95 = (cumy < hpdi * (sum(y)))) %>%
    arrange(x) %>% filter(in95)

  cat(hpdi * 10, "% HPDI: [", min(temp.df$x), ",", max(temp.df$x), "]", sep = "")
}

# Compute 95% HPID
Calc.HPDI(seq(0, 30, 0.01), Posterior.W, 0.95)

```

```

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
## 9.5% HPDI: [9.76,10.14]

```

- (c) Change your prior hyperparameters to represent a *strong* belief that  $\lambda \approx 20$ . This means the mean of the prior should be about 20, but there should be less variance around that mean. Refit your model and summarize the new results. How much have the results changed? And in what ways?

Answer

```

prior.alpha <- 1000
prior.beta  <- 50
post.alpha  <- sum(A1.df$y) + prior.alpha
post.beta   <- nrow(A1.df) + prior.beta

Likelihood  <- dpois(seq(0, 30, 1), lambda.vec[which.max(LogLik.vec)])
Prior.S     <- dgamma(seq(0, 30, 0.01), prior.alpha, prior.beta)
Posterior.S <- dgamma(seq(0, 30, 0.01), post.alpha, post.beta)

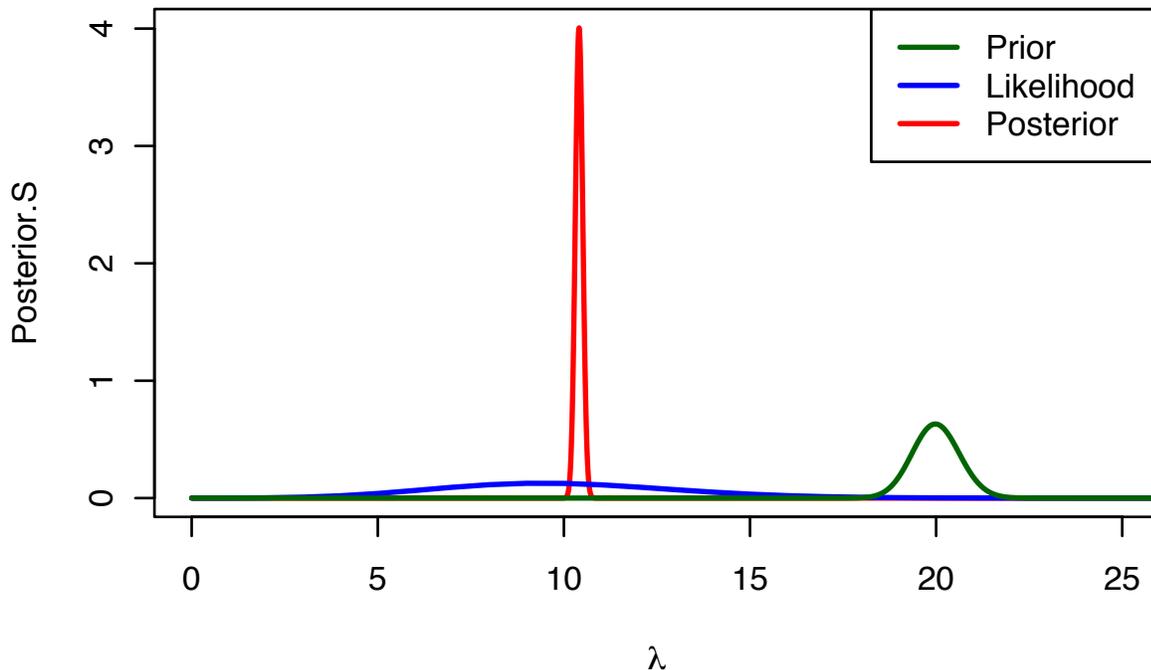
```

```

plot(x = seq(0, 30, 0.01), y = Posterior.S, type = "l",
     lwd = 2, col = "red", xlim = c(0, 25),
     xlab = expression(lambda), main = "Posterior with a strong prior")
lines(x = seq(0, 30, 1), y = Likelihood,
      lwd = 2, col = "blue")
lines(x = seq(0, 30, 0.01), y = Prior.S,
      lwd = 2, col = "darkgreen")
legend("topright", lwd = 2, col = c("darkgreen", "blue", "red"),
      legend = c("Prior", "Likelihood", "Posterior"))

```

Posterior with a strong prior



The EAP, MAP and 95% credible intervals are:

```

Q4.EAP.S <- post.alpha / post.beta
Q4.MAP.S <- (post.alpha - 1) / post.beta

```

```

cat("EAP:", Q4.EAP.S)

```

```

## EAP: 10.40857

```

```

cat("MAP:", Q4.MAP.S)

```

```

## MAP: 10.40762

```

```

Q4.CI.S <- qgamma(c(0.025, 0.975), post.alpha, post.beta)

```

```

cat("95% Credible Intervals: [", Q4.CI.S[1], ", ", Q4.CI.S[2], "]", sep = "")

```

```
## 95% Credible Intervals: [10.21433, 10.60461]
```

- EAP: 10.409
- MAP: 10.408
- 95% credible intervals: [10.214, 10.605]

HPD intervals are:

```
# Compute 95% HPDI  
Calc.HPDI(seq(0, 30, 0.01), Posterior.S, 0.95)
```

```
## 9.5% HPDI: [10.22,10.6]
```

## Answer using Stan

The answers of (b) and (c) is also computed using Stan. For example,

```
library(rstan)  
  
Q4.Model.Code <- "  
data{  
  # Define data.  
  int<lower = 0> N;      # Size of data  
  int y[N];            # Observed data (y)  
  real<lower = 0> alpha; # alpha parameter in prior  
  real<lower = 0> beta;  # beta parameter in prior  
}  
parameters{  
  # Declare the parameter which I want estimate  
  real<lower = 0> lambda;  
}  
model{  
  # y follows Poisson distribution with parameter lambda  
  y ~ poisson(lambda);  
  
  # lambda follows Gamma distribution with parameter alpha and beta  
  lambda ~ gamma(alpha, beta);  
}  
"  
  
# Compile stan model.  
Q4.Model <- stan_model(model_code = Q4.Model.Code)
```

```
## In file included from file34f54aefb753.cpp:8:
```

```
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/
```

```

## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/confi
## # define BOOST_NO_CXX11_RVALUE_REFERENCES
##      ^
## <command line>:6:9: note: previous definition is here
## #define BOOST_NO_CXX11_RVALUE_REFERENCES 1
##      ^
## 1 warning generated.

```

Some warning message was appeared, we can ignore it. Next, I make a list of data and estimate posterior distribution.

```

# Make the data lists
Q4.data.W <- list(N      = nrow(A1.df), # Data size
                 y      = A1.df$y,    # Observed data
                 alpha = 40,          # alpha parameter in weak hyperprior
                 beta  = 2)           # beta parameter in weak hyperprior
Q4.data.S <- list(N      = nrow(A1.df),
                 y      = A1.df$y,
                 alpha = 1000,       # alpha parameter in strong hyperprior
                 beta  = 50)         # beta parameter in strong hyperprior

# Sampling
## Iteration = 2000 * 4 chains (burn-in: 1000)
Q4.Fit1 <- sampling(Q4.Model, data = Q4.data.W,
                  iter = 2000, chains = 4, cores = 4)
Q4.Fit2 <- sampling(Q4.Model, data = Q4.data.S,
                  iter = 2000, chains = 4, cores = 4)

```

```
Q4.Fit1
```

```

## Inference for Stan model: 7e46a7ca12723c4a97257584c029859a.
## 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%
## lambda    9.94    0.00 0.10    9.75    9.88    9.94   10.01   10.14
## lp__  12934.10    0.02 0.69 12932.17 12933.93 12934.37 12934.55 12934.60
##           n_eff Rhat
## lambda  1574    1

```

```

## lp__      1783      1
##
## Samples were drawn using NUTS(diag_e) at Sat Jul 29 10:57:08 2017.
## 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).
Q4.Fit2
## Inference for Stan model: 7e46a7ca12723c4a97257584c029859a.
## 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%
## lambda    10.41     0.00 0.10    10.22    10.35    10.41    10.48    10.61
## lp__     14673.11    0.02 0.69 14671.19 14672.94 14673.37 14673.55 14673.60
##           n_eff Rhat
## lambda    1283      1
## lp__     1279      1
##
## Samples were drawn using NUTS(diag_e) at Sat Jul 29 10:57:11 2017.
## 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).

```

The almost same result of analytical solution is shown.

- (d) The posterior distribution is  $p(\lambda|y) \propto \lambda^{\sum y_i + \alpha - 1} e^{-(n+\beta)\lambda}$ . Explain how the  $\alpha$  parameter in the Gamma prior is like adding event counts to the data and the  $\beta$  parameter is like adding observations. (Target your explanation at a first year graduate student)

Answer

The PMF of Poisson distribution is  $\frac{\lambda^y e^{-\lambda}}{y!}$  and PMF of Gamma distribution is  $\frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}$ . Thus, posterior distribution is:

$$p(\lambda|y) \propto \prod_{i=1}^n \left( \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \right) \cdot \left( \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \right) \quad (8)$$

We can rewrite right hand of Equation 8 as follow:

$$p(\lambda|y) \propto \left( \prod_{i=1}^n \frac{1}{y_i!} \right) \lambda^{\sum y_i} e^{-n\lambda} \cdot \left( \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \right)$$

Next, elements that do not contain the parameter ( $\lambda$ ) is ignorable, thus:

$$\begin{aligned}
p(\lambda|y) &\propto \lambda^{\sum y_i} e^{-n\lambda} \cdot \lambda^{\alpha-1} e^{-\beta\lambda} \\
&\propto \lambda^{(\sum y_i)+\alpha-1} e^{-n\lambda-b\lambda} \\
&\propto \lambda^{(\sum y_i)+\alpha-1} e^{-(n+b)\lambda}
\end{aligned} \tag{9}$$

Replace  $(\sum y_i) + \alpha$  to  $\alpha^*$  and  $n + b$  to  $\beta^2$ , Equation 9 can be expressed as  $\mathcal{Gamma}(\alpha^*, \beta^*) = \mathcal{Gamma}((\sum y_i) + \alpha, n + b)$

## Bonus

Consider the following model:

$$\begin{aligned} y|\pi &\sim \text{Binomial}(\pi) \\ \pi|a, b &\sim \text{Beta}(a, b) \end{aligned}$$

- Show that the posterior predictive distribution,  $p(y^*|y, \pi, a, b)$ , follows the Beta-Binomial Distribution.

Answer

The posterior predictive distribution is

$$p(y^*|n^*, y, \pi, a, b) = \int_0^1 p(y^*|n^*, \pi) p(\pi|n, y, a, b) d\pi, \quad (10)$$

where  $n^*$  is a number of trials of predictive observation.

The Beta distribution is the conjugate of Binomial distribution, thus, posterior distribution of  $\pi$  is  $\text{Beta}(a^*, b^*)$  and  $a^* = a + y, b^* = b + n - y$ . Equation 10 can be rewritten as:

$$\begin{aligned} p(y^*|y, \pi, a, b) &= \int_0^1 \underbrace{\binom{n}{k} \pi^{y^*} (1 - \pi)^{n^* - y^*}}_{\text{Binomial}(y^*|n^*, \pi)} \underbrace{\frac{1}{\text{B}(a^*, b^*)} \pi^{(a^* - 1)} (1 - \pi)^{(b^* + 1)}}_{\text{Beta}(a^*, b^*)} d\pi \\ &= \binom{n}{k} \frac{1}{\text{B}(a^*, b^*)} \int_0^1 \pi^{y^*} (1 - \pi)^{n^* - y^*} \pi^{(a^* - 1)} (1 - \pi)^{(b^* - 1)} d\pi \\ &= \binom{n}{k} \frac{1}{\text{B}(a^*, b^*)} \int_0^1 \pi^{(a^* + y^* - 1)} (1 - \pi)^{(b^* + n^* - y^* - 1)} d\pi. \end{aligned} \quad (11)$$

Beta function,  $\text{B}(a, b)$ , is defined as

$$\text{B}(a, b) = \int_0^1 t^{a-1} (1 - t)^{b-1} dt. \quad (12)$$

Thus,

$$\begin{aligned} p(y^*|y, \pi, a, b) &= \binom{n}{k} \frac{1}{\text{B}(a^*, b^*)} \underbrace{\int_0^1 \pi^{(a^* + y^* - 1)} (1 - \pi)^{(b^* + n^* - y^* - 1)} d\pi}_{\text{B}(a^* + y^*, b^* + n^* + y^*)} \\ &= \binom{n}{k} \frac{\text{B}(a^* + y^*, b^* + n^* - y^*)}{\text{B}(a^*, b^*)} \end{aligned} \quad (13)$$

The form of Equation 13 is consistency in the form of  $\text{BetaBin}(y^*|n^*, a^*, b^*)$ .

- Imagine that you cannot directly sample from the Beta-Binomial distribution. Define a sampling scheme that would allow you to draw predicted data  $y^*$  conditional on the observed data and model setup above.

Answer

To draw sample from Beta-Binomial distribution, we can use a simulation as follow steps:

Step 1. Draw a sample,  $\pi^*$ , from  $\text{Beta}(a + y, b + n - y)$ .

Step 2. Draw a sample,  $y^*$ , from  $\text{Binomial}(n^*, \pi^*)$ .

Step 3. Repeat

These steps can be written in R as follow:

```
Sampling.BB <- function(n, y, a, b, new.n, iter = 1000){
  # Arguments
  # n: number of trials in observed data
  # y: number of successes in observed data
  # a: alpha parameter in prior distribution
  # b: beta parameter in prior distribution
  # n: number of trials in predictive observation
  # iter: number of iterations

  # Preparation of result vector.
  result.vec <- rep(NA, iter)

  for (i in 1:iter) {
    # Draw a sample (pi*) from the Beta distribution
    temp.pi <- rbeta(1, a + y, b + n - y)
    # Draw a sample (y*) from the Binomial distribution with
    # the parameter drawn from the Beta distribution (p*).
    new.y <- rbinom(1, new.n, temp.pi)

    # Store the new data (y*) into the result vector.
    result.vec[i] <- new.y
  }

  # Return the vector.
  return(result.vec)
}

# Simulation
# n = 10, y = 3, a = 2, b = 5, new.n = 100
Sample.BB <- Sampling.BB(n = 10, y = 3, a = 2, b = 5, new.n = 100,
```

```

iter = 10000)

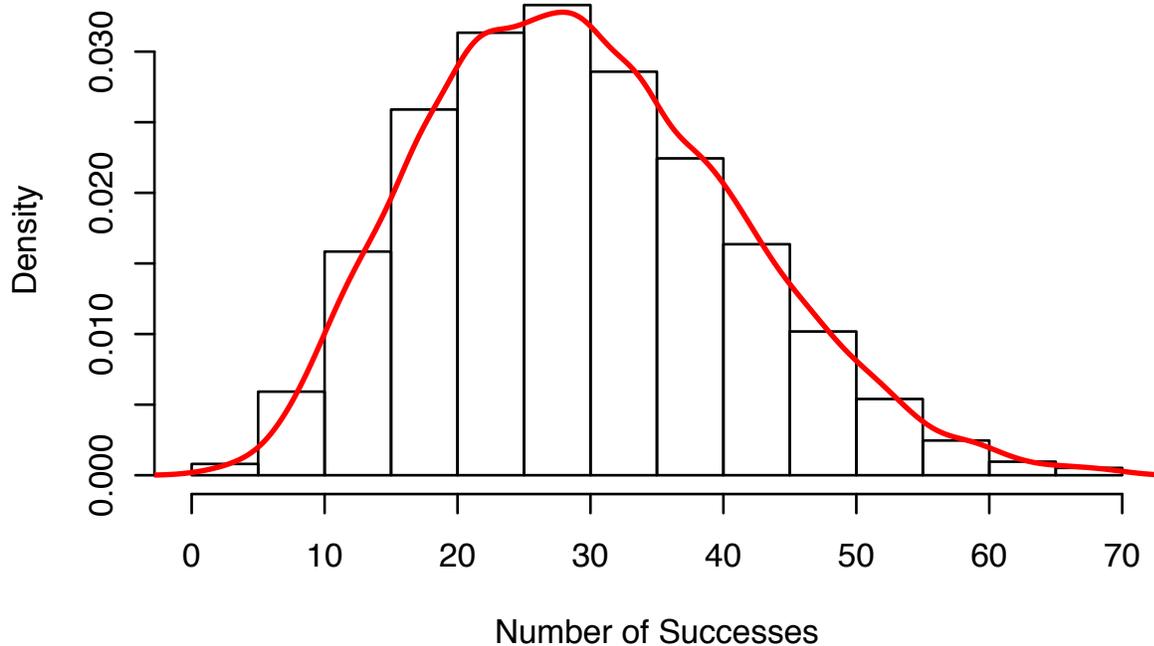
# Check the descriptive statistics of the sample
summary(Sample.BB)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00  21.00  29.00  29.45  37.00  70.00

hist(Sample.BB, freq = FALSE, xlab = "Number of Successes",
      main = "Sampled data from Beta-Binomial distribution")
lines(density(Sample.BB), lwd = 2, col = "red")

```

Sampled data from Beta-Binomial distribution



The result from the code above can be replicated using Stan as below

```

B.code <- "
data{
  # Define the data and parameter setting
  int<lower = 0> N;
  int<lower = 0> y;
  real<lower = 0> a;
  real<lower = 0> b;
  int<lower = 0> new_N;
}
parameters{
  # Paramters to be estimated.

```

```

    real<lower = 0, upper = 1> pi;
}
model{
  # y follows the binomial distribution with N and pi
  y ~ binomial(N, pi);

  # pi follows the beta distribution with a and b
  pi ~ beta(a, b);
}
generated quantities{
  # Declare what I will draw
  int<lower = 0> new_y;

  # New y is generated the estimated pi
  new_y = binomial_rng(new_N, pi);
}
"

```

*# Model compiling*

```
B.model <- stan_model(model_code = B.code)
```

```

## In file included from file3e35612021d8.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/confi
## # define BOOST_NO_CXX11_RVALUE_REFERENCES
##      ^
## <command line>:6:9: note: previous definition is here
## #define BOOST_NO_CXX11_RVALUE_REFERENCES 1
##      ^
## 1 warning generated.

```

```
B.data <- list(N = 10, y = 3, a = 2, b = 5, new_N = 100)
```

*# Sampling*

```
B.sample <- sampling(B.model, B.data,
                    iter = 2000, chains = 4, cores = 4)
```

*# Extract sampled new\_y*

```
Sampled.y <- extract(B.sample, pars = "new_y")$new_y
```

```
# Check the descriptive statistics of the sample  
summary(Sampled.y)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##      1.00  21.00   29.00   29.67  37.00   73.00
```

```
# Plot the distribution
```

```
hist(Sampled.y, freq = FALSE, xlab = "Number of Successes",  
      main = "Sampled data from Beta-Binomial distribution (Stan ver.)")  
lines(density(Sampled.y), lwd = 2, col = "red")
```

Sampled data from Beta-Binomial distribution (Stan ver.)



The result from the sampling using Stan is approximately equivalent to that of my-own simulation function.