STAT 450: Bayesian Statistics - Homework 7

Problem 1: Beta-Binomial Model

(a)

```
a <- 15
b <- 85
```

(b)

```
mtext <- "
model{
  y ~ dbinom(theta, n)
  theta ~ dbeta(a, b)
}
"

#step 2: data and priors
dat <- list(y = 30, n = 90, a = 15, b = 85)

start_val <- list(
  list("theta" = 0.5,
                  ".RNG.name" = "base::Wichmann-Hill",
                  ".RNG.seed" = 45071),
  list("theta" = 0.7,
                    ".RNG.name" = "base::Wichmann-Hill",
                    ".RNG.seed" = 45072)
)

out <- run.jags(model = mtext,
                monitor = "theta",
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:39 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
```

```
##    Observed stochastic nodes: 1
##    Unobserved stochastic nodes: 1
##    Total graph size: 5
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
## +++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************* 100%
## . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************* 100%
## . . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 1 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]
thetas <- c(chain1, chain2)
```

Mean, Standard Deviation, and 95% credible interval

```
mean(thetas)
```

```
## [1] 0.2367195
```

```
sd(thetas)
```

```
## [1] 0.03033443
```

```
quantile(thetas, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.1808150 0.2992802
```

(c)

```
mtext <- "
model{
  y ~ dbinom(theta, n)
  theta ~ dbeta(a, b)
}
"

dat <- list(y = 30, n = 90, a = 1/2, b = 1/2)   #Beta(1/2,1/2) is the JP for theta

start_val <- list(
  list("theta" = 0.5,
```

```
                  ".RNG.name" = "base::Wichmann-Hill",
                  ".RNG.seed" = 45071),
  list("theta" = 0.7,
                    ".RNG.name" = "base::Wichmann-Hill",
                    ".RNG.seed" = 45072)
)

out <- run.jags(model = mtext,
                monitor = "theta",
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:41 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1
##    Unobserved stochastic nodes: 1
##    Total graph size: 5
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
## +++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************* 100%
## . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************* 100%
## . . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 1 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]
thetas <- c(chain1, chain2)

mean(thetas)
```

```
## [1] 0.335282
```

```
sd(thetas)
```

```
## [1] 0.0485728
```

```
quantile(thetas, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.2440569 0.4342364
```

(d)

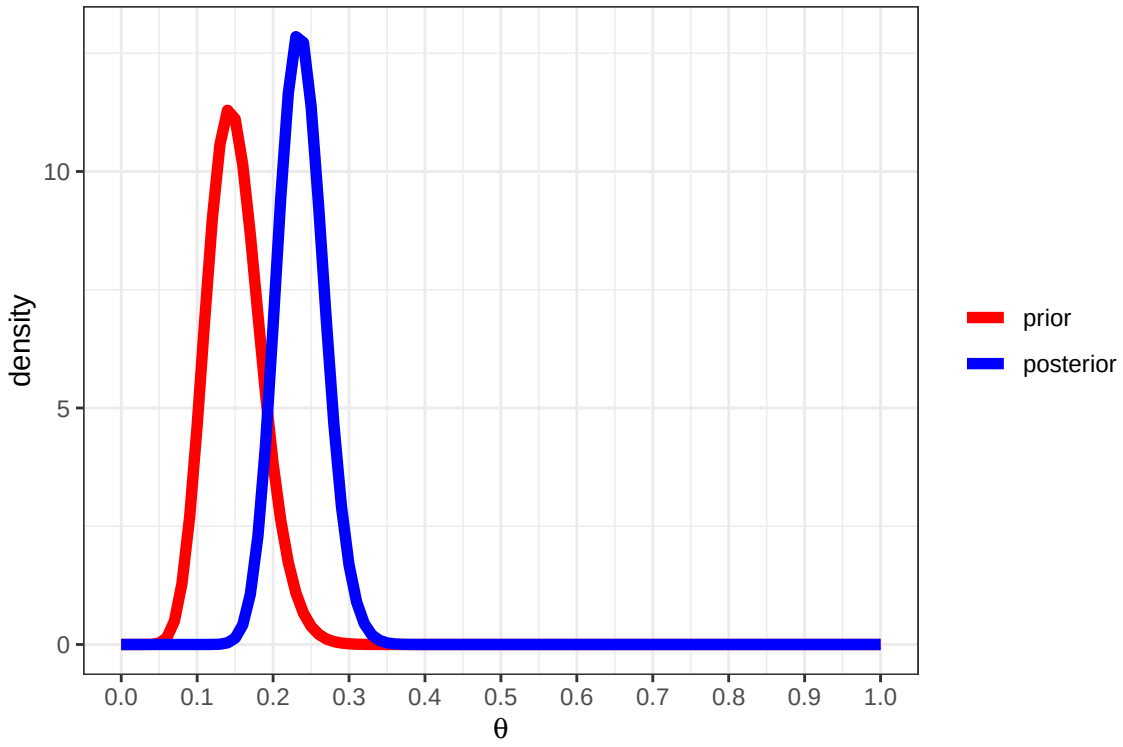
```
xrange <- c(0, 1)
df <- data.frame(xrange)
y <- 30
n <- 90

# specify prior parameters
a <- 15
b <- 85

# specify posterior parameters
a_post <- a + y
b_post <- b + n - y

ggplot(data = df, aes(x = xrange)) +
  geom_function(fun = function(x) dbeta(x, a, b), aes(color = "prior"), lwd=2) +   # prior
  geom_function(fun = function(x) dbeta(x, a_post, b_post), aes(color = "posterior"), lwd=2) +   # post
  scale_x_continuous(breaks = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)) +
  scale_colour_manual(breaks = c("prior", "posterior"),
                      values = c("red", "blue")) +
  theme_bw() +
  labs(x = expression(theta), y = "density") +
  theme(legend.title = element_blank())
```

Beta Priors



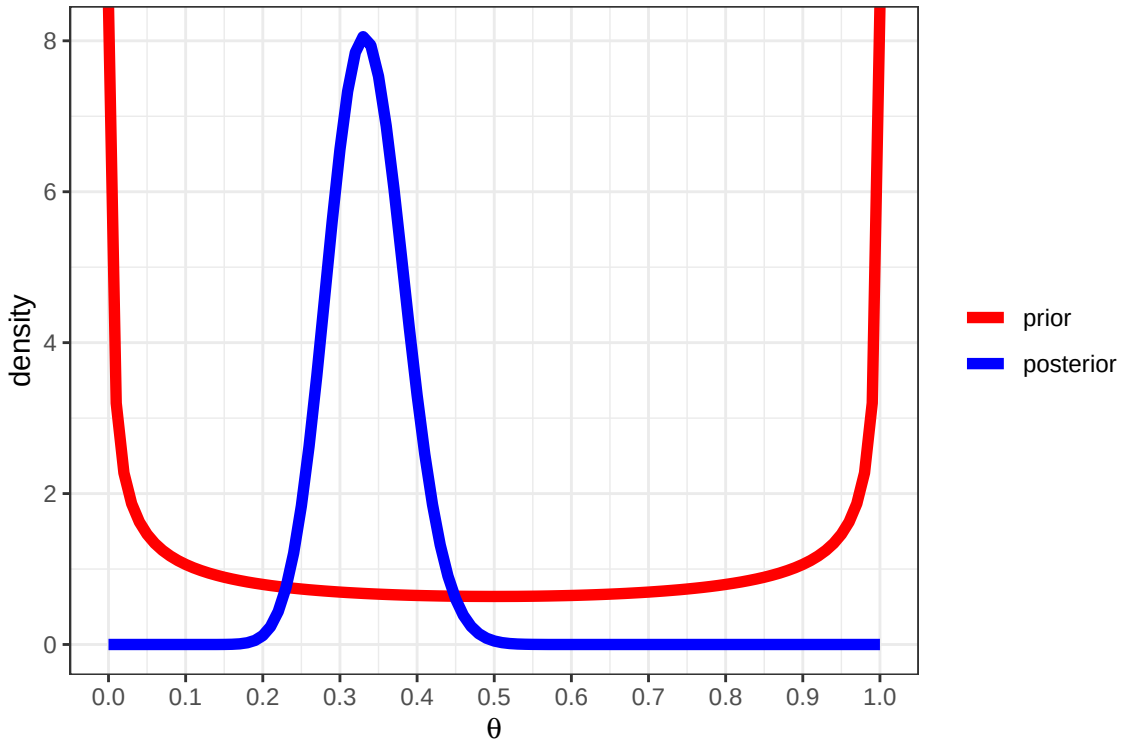

```
xrange <- c(0, 1)
df <- data.frame(xrange)
y <- 30
n <- 90

# specify prior parameters
a <- 0.5
b <- 0.5

# specify posterior parameters
a_post <- a + y
b_post <- b + n - y

ggplot(data = df, aes(x = xrange)) +
  geom_function(fun = function(x) dbeta(x, a, b), aes(color = "prior"), lwd=2) +   # prior
  geom_function(fun = function(x) dbeta(x, a_post, b_post), aes(color = "posterior"), lwd=2) +   # post
  scale_x_continuous(breaks = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)) +
  scale_colour_manual(breaks = c("prior", "posterior"),
                      values = c("red", "blue")) +
  theme_bw() +
  labs(x = expression(theta), y = "density") +
  theme(legend.title = element_blank())
```

Jeffreys Priors



Beta priors: more weight from the prior was factored in the calculations of the posteriors, which means the model for sampling was able to rely on both the priors and the data. The posteriors mean reflects this: it is within the range of the prior mean and the data mean.

Jeffreys prior: The 95% credible interval for the JP was wider than the 95% credible interval of the Beta prior because the JP did not provide much information for the posterior. The posterior mean was closer to the data mean because the prior does not have much information and the model and posterior calculations relied more on the data.

Problem 2: Gamma-Poisson Model

```
goals <- c(3, 2, 2, 3, 5, 4, 3, 4, 5, 5, 5, 3, 6, 4, 6, 4, 5, 5, 3, 2, 2, 3, 5, 1, 1, 4, 4, 5, 5, 3, 3,
```

(a)

```
mtext <- "
model{
for(i in 1:n){
    y[i] ~ dpois(lambda)
  }
  lambda ~ dgamma(a,b)
}
"

dat <- list(y = goals, n = length(goals), a = 1, b = 1)

start_val <- list(
  list("lambda" = 2,
```

```
                  ".RNG.name" = "base::Wichmann-Hill",
                  ".RNG.seed" = 45071),
  list("lambda" = 5,
                    ".RNG.name" = "base::Wichmann-Hill",
                    ".RNG.seed" = 45072)
)


out <- run.jags(model = mtext,
                monitor = "lambda",
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000,
                autocorr.lags = 10)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:43 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 69
##    Unobserved stochastic nodes: 1
##    Total graph size: 73
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************** 100%
## . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************** 100%
## . . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 1 variables....
## Finished running the simulation
```

(b)

Posterior mean, variance, and 95% credible interval

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]
lambdas <- c(chain1, chain2)

mean(lambdas)
```

```
## [1] 3.441382
```

```
var(lambdas)
```

```
## [1] 0.04936461
```

```
quantile(lambdas, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 3.019669 3.888021
```

(c)

```
mean(lambdas > 3)
```

```
## [1] 0.9806
```

Problem 3: Times Between Traffic Accidents

```
y <- c(1.5, 15, 60.3, 30.5, 2.8, 56.4, 27, 6.4, 110.7, 25.4)
```

(b)

```
mtext <- "
model{
for(i in 1:n){
    y[i] ~ dexp(lambda)
  }
  lambda ~ dgamma(a+n,sumy+b)
  ytilde ~ dexp(lambda)
}
"

dat <- list(y = y, n = length(y), a = 1, b = 1, sumy = sum(y))

start_val <- list(
  list("lambda" = 2, "ytilde"= 14,
                  ".RNG.name" = "base::Wichmann-Hill",
                  ".RNG.seed" = 45071),
  list("lambda" = 5, "ytilde"= 19,
                    ".RNG.name" = "base::Wichmann-Hill",
                    ".RNG.seed" = 45072)
)


out <- run.jags(model = mtext,
                monitor = c("lambda","ytilde"),
```

```
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000,
                autocorr.lags = 10)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:43 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 10
##    Unobserved stochastic nodes: 2
##    Total graph size: 18
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************** 100%
## . . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************** 100%
## . . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 2 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]

lambdas <- c(chain1[,1], chain2[,1])
ytildes <- c(chain1[,2], chain2[,2])
```

(c)

Lambdas summaries (mean, standard deviation, 95% credible interval):

```
mean(lambdas)
```

```
## [1] 0.03130536
```

```
sd(lambdas)
```

```
## [1] 0.006848482
```

```r
quantile(lambdas, c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.01940817 0.04609314
```

(d)

```r
mean(ytildes)
```

```
## [1] 33.34339
```

On average, it is expected that there is a traffic accident every 33.34 minutes.

(e)

I replace the prior lambda with the posterior lambda.

Problem 4: NFL Concussions Data

```r
mtext <- "
model
{
  for(i in 1:4){
    y[i] ~ dpois(n*lambda[i])
    lambda[i] ~ dgamma(1, gamma)
  }
  gamma ~ dgamma(a,b)
}
"

dat <- list(y = c(171, 152, 123, 199), n = 256, a = 0.1, b = 0.1)

start_val <- list(list("lambda" = c(171/256, 152/256, 123/256, 199/256), "gamma" = 1.5,
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45071),
                  list("lambda" = c(0.8,0.8,0.8,0.8), "gamma" = 2.9,
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45072))

out <- run.jags(model = mtext,
                monitor = c("lambda", "gamma"),
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000,
                autocorr.lags = 10)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:44 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
```

```
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 4
##    Unobserved stochastic nodes: 5
##    Total graph size: 17
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************** 100%
## . . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************** 100%
## . . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 5 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]

lambdas <- c(chain1[,1], chain2[,1])
gammas <- c(chain1[,2], chain2[,2])
```

Lambdas summaries (mean, median, 95% credible interval):

```
mean(lambdas)
```

```
## [1] 0.667347
```

```
median(lambdas)
```

```
## [1] 0.665664
```

```
quantile(lambdas, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5711323 0.7702947
```

Gammas summaries (mean, median, 95% credible interval):

```
mean(gammas)
```

```
## [1] 0.5947882
```

```
median(gammas)
```

```
## [1] 0.5934285
```

```r
quantile(gammas, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5047504 0.6930617
```

Problem 5: ALS Data With a Common Rate

```r
mtext <- "
model
{
  for(i in 1:2){
    y[i] ~ dpois(theta*m[i])
  }
  theta ~ dnorm(2,5)
}
"
dat <- list(y = c(5, 8), m = c(1,2))

start_val <- list(list("theta" = 1,
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45071),
                  list("theta" = 0.8,
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45072)
                  )

out <- run.jags(model = mtext,
                monitor = "theta",
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000,
                autocorr.lags = 10)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:46 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 1
##    Total graph size: 9
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adapting 1000
## -------------------------------------------------| 1000
```

```
## ++++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 4000
## --------------------------------------------------| 4000
## ************************************************** 100%
## . . Updating 10000
## --------------------------------------------------| 10000
## ************************************************** 100%
## . . . . . Updating 0
## . Deleting model
## .
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 1 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]
thetas <- c(chain1, chain2)
```

Thetas summaries (mean, median, 95% credible interval):

```
mean(thetas)
```

```
## [1] 2.47828
```

```
median(thetas)
```

```
## [1] 2.473435
```

```
quantile(thetas, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.778178 3.213233
```

Problem 6: ALS Data With Different Rates

(b)

```
mtext <- "
model
{
  for (i in 1:2)
  {
    y[i] ~ dpois(theta[i]*m[i])
    theta[i] ~ dgamma(yth[i]+a[i],m[i]+b[i])
  }
}"
dat <- list(y = c(5, 8), m = c(1, 2), a = c(1,1), b = c(1,1), yth=c(5,8))



start_val <- list(list("theta" = c(0.6,0.8),
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45071),
```

```
                  list("theta" = c(1,1.5),
                       ".RNG.name" = "base::Wichmann-Hill",
                       ".RNG.seed" = 45072))

out <- run.jags(model = mtext,
                monitor = "theta",
                data = dat,
                n.chains = 2,
                inits = start_val,
                sample = 10000)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.2 on Mon May 13 15:17:47 2024
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 2
##    Total graph size: 18
## . Reading parameter file inits1.txt
## . Reading parameter file inits2.txt
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 4000
## -------------------------------------------------| 4000
## ************************************************** 100%
## . . Updating 10000
## -------------------------------------------------| 10000
## ************************************************** 100%
## . . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Calculating the Gelman-Rubin statistic for 2 variables....
## Finished running the simulation
```

```
chain1 <- out$mcmc[[1]]
chain2 <- out$mcmc[[2]]
theta1 <- c(chain1[,1], chain2[,1])
theta2 <- c(chain1[,2], chain2[,2])
```

(c)

```
mean(theta1 > theta2)
```

```
## [1] 0.56395
```

(d)

```
quantile(theta1 - theta2, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -2.339447  3.125230
```

On average, the death rate from ASL in city 1 is between 3% fewer and 3% more than that of city 2. 0 is in the interval, so there is also a chance that the death rates from ASL of the 2 cities are not so different.
