Online sellers worksheet

Joachim Vandekerckhove

Data

The data are online ratings data obtained from 3Blue1Brown (`https://www.youtube.com/watch?v=8idr1WZ1A7Q`). They are very small, so we can enter them manually.

```
k <- c(10, 48, 186)
n <- c(10, 50, 200)
nSellers <- length(k)
```

JAGS Model

The JAGS model is a direct translation of our model assumptions.

```
modelString <- "
model {
  for (i in 1:nSellers) {
    k[i] ~ dbin(theta[i], n[i])
    theta[i] ~ dunif(0, 1)
  }
}
"
jags_file <- "onlineSellers_solution.jags"
writeLines(modelString, con = jags_file)

dataList <- list(k = k, n = n, nSellers = nSellers)
```

MCMC Sampling

To set up the MCMC sampler, begin by listing the parameter(s) of interest and writing an initial values generating function for them.

```r
# Parameters to monitor
params <- c("theta")

# Initial values
inits <- function() {
  list(theta = runif(nSellers))
}
```

Compile and start adaptation for 1000 samples (per chain).

```r
# JAGS model
model <- jags.model(jags_file,
                    data     = dataList,
                    inits    =    inits,
                    n.chains =        8 ,
                    n.adapt  =     1000 )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 3
##    Unobserved stochastic nodes: 3
##    Total graph size: 12
##
## Initializing model
```

Do 1000 burn-in and then retain 5000 samples (per chain).

```r
# Burn-in
update(model, n.iter = 1000)

# Sampling
samples <- coda.samples(model,
                        variable.names = params,
                        n.iter         =  5000 )

# Convert to matrix for easier handling
samples_mx <- as.matrix(samples)
```

Inspect the Results

The potential scale reduction factor is a standard diagnostic, it should be close to 1 and ideally less than 1.1.

```
print(gelman.diag(samples))
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## theta[1]          1       1.01
## theta[2]          1       1.00
## theta[3]          1       1.00
##
## Multivariate psrf
##
## 1
```

Print some summary statistics.

```r
summary(samples)[1]$
  statistics[, c("Mean", "SD", "Time-series SE")] %>%
  print()
```

```
##                  Mean         SD Time-series SE
## theta[1] 0.9172730 0.07650044   0.0008055408
## theta[2] 0.9425132 0.03173722   0.0002497739
## theta[3] 0.9259358 0.01844607   0.0001210348
```

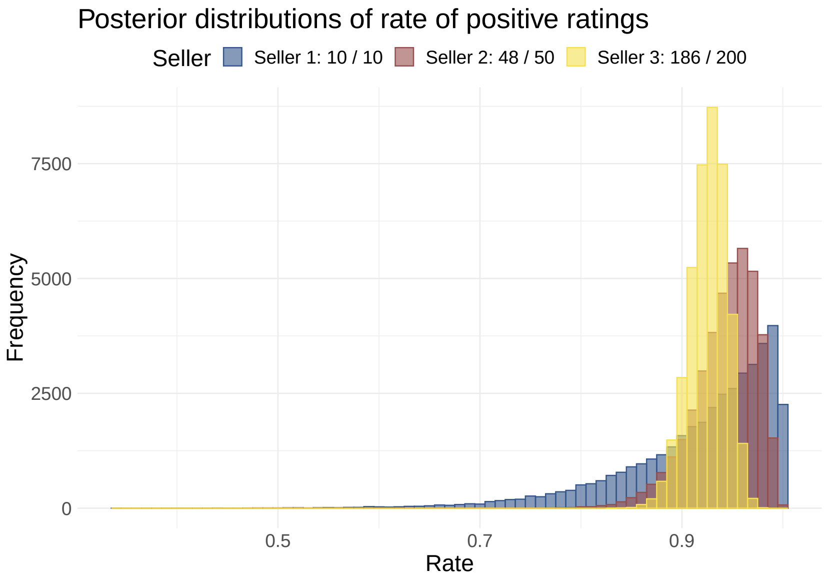
Visualize the posterior samples.

```r
lineWidth <- 3
sellerLabel <- c("Seller 1: 10 / 10",
                 "Seller 2: 48 / 50",
                 "Seller 3: 186 / 200")

# Create data for plotting histograms
histogram_data <- lapply(1:nSellers, function(i) {
  data.frame(
    x = samples_mx[, paste0("theta[", i, "]")],
    Seller = sellerLabel[i]
  )
})

histogram_data <- bind_rows(histogram_data)
```

Plot the histograms.

```r
sellerColors <- c("#34568B", "#964F4C", "#F5E050")
d1 <- ggplot(histogram_data, aes(x = x, fill = Seller,
                                 color = Seller)) +
  geom_histogram(binwidth = 0.01, alpha = 0.6,
                 position = "identity") +
  scale_fill_manual(values = sellerColors) +
  scale_color_manual(values = sellerColors) +
  labs(x = "Rate", y = "Frequency",
       title = paste0("Posterior distributions of ",
                      "rate of positive ratings")) +
  theme_minimal() +
  theme(
    text = element_text(size = 20),
    legend.position = "top"
  )
```

```r
print(d1)
```


Posterior distributions of rate of positive ratings
Seller
Seller 1: 10 / 10
Seller 2: 48 / 50
Seller 3: 186 / 200
Frequency
7500
5000
2500
0
0.5
0.7
0.9
Rate