HW 4

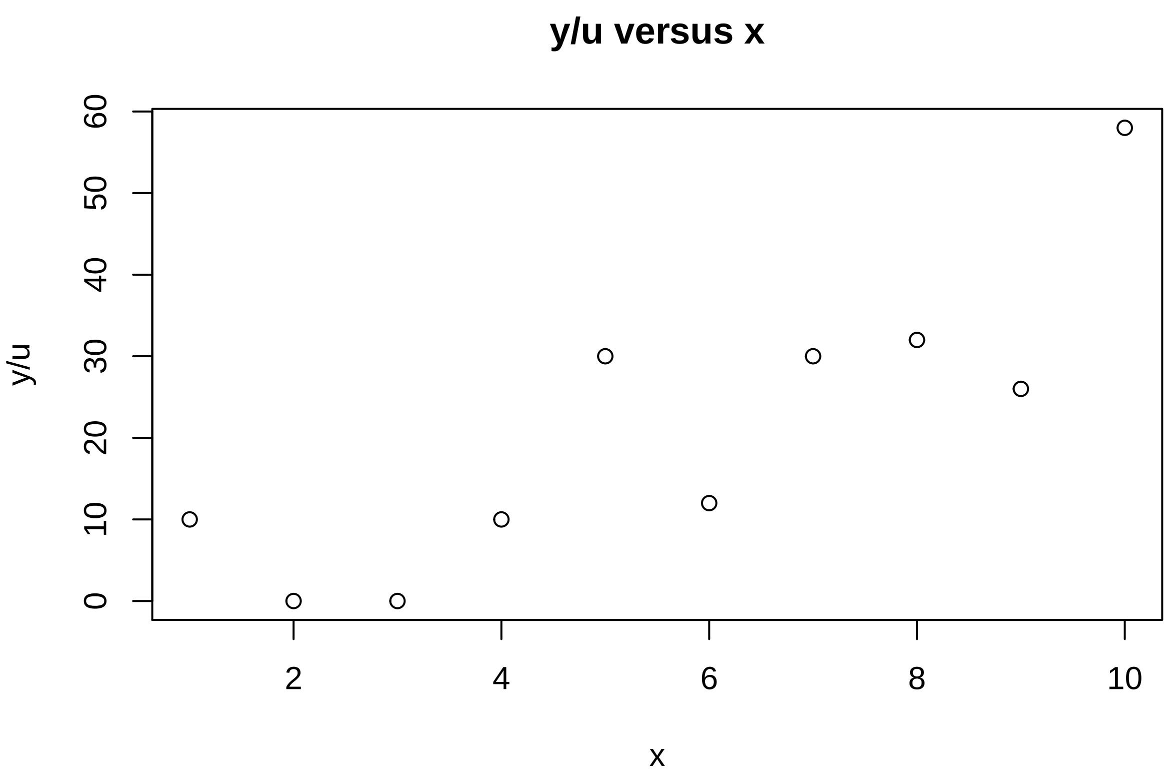
Problem 1

(1)

```r
set.seed(0)

n <- 10
x <- seq(1, 10, 1)
u <- rep(c(.1, .5), c(5, 5))
a <- 1.
b <- .3

y <- c()
for (i in 1:n) {
  y <- c(y, rpois(1, lambda=u[i]*exp(a+b*x[i])))
}
y
```

```
##  [1]  1  0  0  1  3  6 15 16 13 29
```

```r
plot(x, y/u, main="y/u versus x")
```



(2)

$$
\begin{aligned}
p(\alpha, \beta|y, u, x) &\propto p(\alpha, \beta)p(y|\alpha, \beta, u, x) \\
&\propto \prod_{i=1}^{n} \frac{(\lambda_i)^{y_i} e^{\lambda_i}}{y_i!},
\end{aligned}
$$

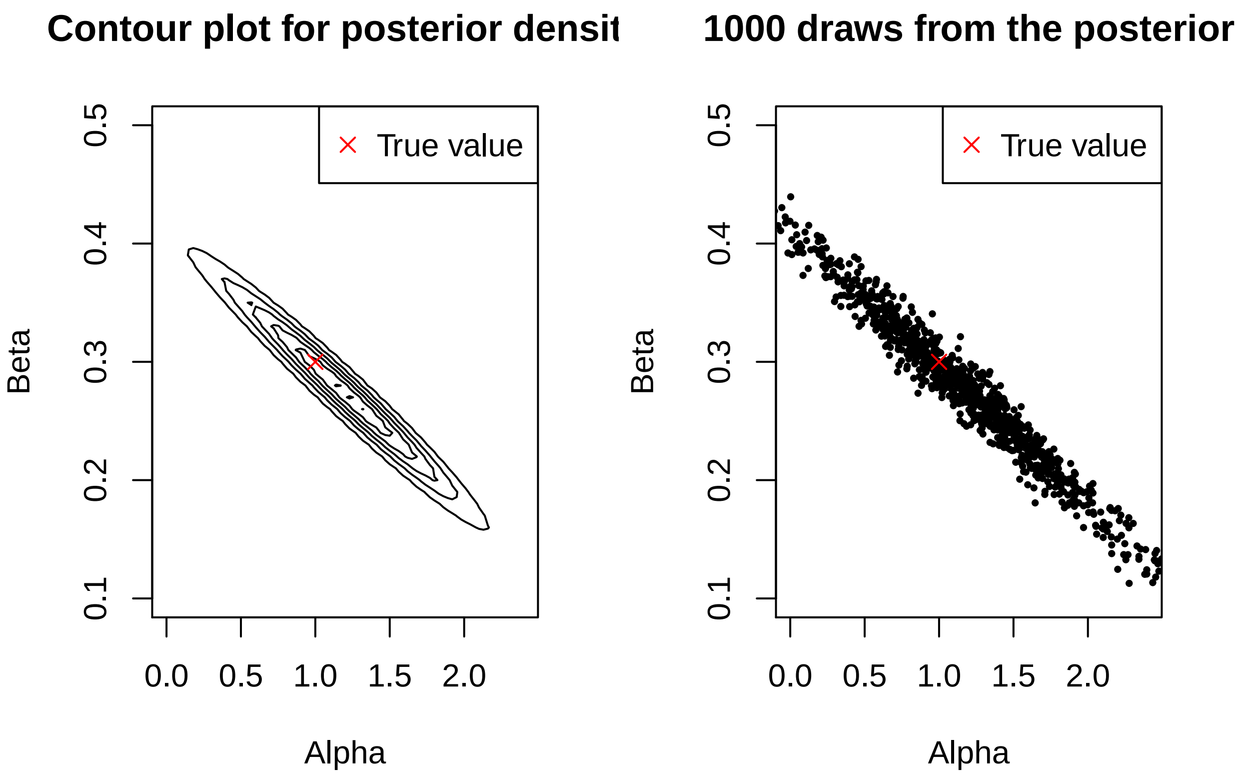
where $\lambda_i = u_i \exp(\alpha + \beta x_i)$.

(3)

```r
log_posterior <- function(a, b) {
  # x,u,y are given and fixed
  lambda <- u*exp(a+b*x)
  sum(log(lambda^y * exp(-lambda) / factorial(y)))
}

iter <- 1e5
res <- matrix(nrow=iter+1, ncol=2)
res[1, ] <- c(1., .3) # initial value
for (i in 1:iter) {
  temp_alpha <- runif(1, res[i, 1]-.3, res[i, 1]+.3) # proposal
  log_ratio <- log_posterior(temp_alpha, res[i, 2]) -
    log_posterior(res[i, 1], res[i, 2])
  temp_log_U <- log(runif(1))
  if (temp_log_U <= log_ratio) {
    res[i+1, 1] <- temp_alpha
  } else {
    res[i+1, 1] <- res[i, 1]
  }

  temp_beta <- runif(1, res[i, 2]-.05, res[i, 2]+.05) # proposal
  log_ratio <- log_posterior(res[i+1, 1], temp_beta) -
    log_posterior(res[i+1, 1], res[i, 2])
  temp_log_U <- log(runif(1))
  if (temp_log_U <= log_ratio) {
    res[i+1, 2] <- temp_beta
  } else {
    res[i+1, 2] <- res[i, 2]
  }
}
res <- tail(res, -20000)
```

```r
library(ggplot2)
xgrid <- seq(0, 2.4, .03)
ygrid <- seq(0.1, .5, .01)
z <- matrix(nrow=length(xgrid), ncol=length(ygrid))
for (i in 1:length(xgrid)) {
  for (j in 1:length(ygrid)) {
    z[i, j] <- exp(log_posterior(xgrid[i], ygrid[j]))
  }
}
par(mfrow=c(1,2))
contour(x=xgrid, y=ygrid, z=z, drawlabels=F,
```

```
        main='Contour plot for posterior density',
        xlab='Alpha', ylab='Beta')
points(1, .3, col='red', pch=4)
legend("topright", legend='True value', pch=4, col='red')
plt_df <- res[sample(nrow(res), 1e3), ]
plot(plt_df[, 1], plt_df[, 2],
     xlim=c(0, 2.4), ylim=c(0.1, .5), pch=16, cex=.5,
     main='1000 draws from the posterior',
     xlab='Alpha', ylab='Beta')
points(1, .3, col='red', pch=4)
legend("topright", legend='True value', pch=4, col='red')
```



Problem 2 (Exercise 5.13)

(a)

Suppose

$$y_j \sim \text{Bin}(n_j, \theta_j),$$

with parameter n_j known. The parameter θ_j are assumed to be independent samples from a beta distribution:

$$\theta_j \sim \text{Beta}(\alpha, \beta),$$

where α and β are assigned with a noninformative prior, that is, $p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}$.

The posterior is

$$\begin{aligned} p(\theta, \alpha, \beta | y) &\propto p(\alpha, \beta) p(\theta|\alpha, \beta) p(y|\theta, \alpha, \beta) \\ &\propto (\alpha + \beta)^{-5/2} \prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1} (1 - \theta_j)^{\beta-1} \prod_{j=1}^{10} \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j}. \end{aligned}$$

For θ along the posterior is

$$p(\theta|\alpha,\beta,y) \propto \prod_{j=1}^{10} \theta_j^{\alpha+y_j-1}(1-\theta_j)^{\beta+n_j-y_j-1},$$

since it is recognized as independent beta distribution for each j, we know that the constant that we ignored is just $\frac{\Gamma(\alpha+\beta+n_j)}{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}$ for each j.

(b) & (c)

The marginal posterior for α and β is

$$\begin{aligned} p(\alpha,\beta|y) &= \frac{p(\theta,\alpha,\beta|y)}{p(\theta|\alpha,\beta,y)} \\ &\propto (\alpha+\beta)^{-5/2}\prod_{j=1}^{10}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)}. \end{aligned}$$

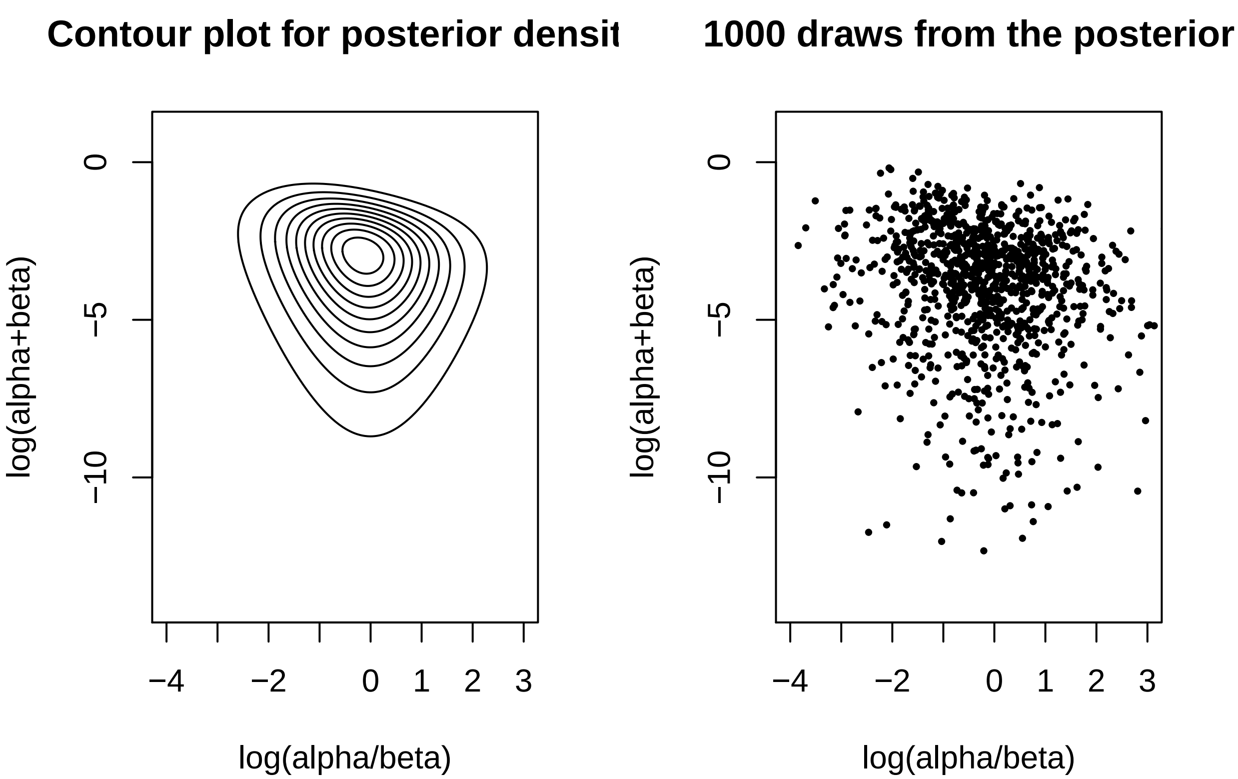
In terms of $(\log(\alpha/\beta), \log(\alpha+\beta))$ parametrization, the marginal posterior can be obtained by multiplying the Jacobian of function $f(\eta,\phi) : (\eta,\phi) \mapsto \left(e^{\phi} - \frac{e^{\phi}}{e^{\eta}+1}, \frac{e^{\phi}}{e^{\eta}+1}\right)$ which is the inverse function of $(\alpha,\beta) \mapsto \left(\log(\frac{\alpha}{\beta}), \log(\alpha,\beta)\right)$:

$$\begin{aligned} p\left(\log(\alpha/\beta), \log(\alpha+\beta)|y\right) &\propto p(\alpha,\beta|y)|\det(J)| \\ &\propto p(\alpha,\beta|y)\,\alpha\beta. \end{aligned}$$

```
y <- c(16,  9, 10, 13, 19, 20, 18, 17, 35, 55)
n <- c(58, 90, 48, 57, 103, 57, 86, 112, 273, 64)


log_posterior <- function(eta, phi) {
  b <- exp(phi) / (exp(eta)+1)
  a <- exp(phi) - b
  log(a) + log(b) - 5/2*log(a+b) + sum(lgamma(a+b)) + sum(lgamma(a+y)) + sum(lgamma(b+n-y)) -
    sum(lgamma(a)) - sum(lgamma(b)) - sum(lgamma(a+b+n))
}

iter <- 1e5
res <- matrix(nrow=iter+1, ncol=2)
res[1, ] <- c(0, 0) # initial value
for (i in 1:iter) {
  temp_alpha <- runif(1, res[i, 1]-1, res[i, 1]+1) # proposal
  log_ratio <- log_posterior(temp_alpha, res[i, 2]) -
    log_posterior(res[i, 1], res[i, 2])
  temp_log_U <- log(runif(1))
  if (temp_log_U <= log_ratio) {
    res[i+1, 1] <- temp_alpha
  } else {
    res[i+1, 1] <- res[i, 1]
  }

  temp_beta <- runif(1, res[i, 2]-1, res[i, 2]+1) # proposal
  log_ratio <- log_posterior(res[i+1, 1], temp_beta) -
    log_posterior(res[i+1, 1], res[i, 2])
```

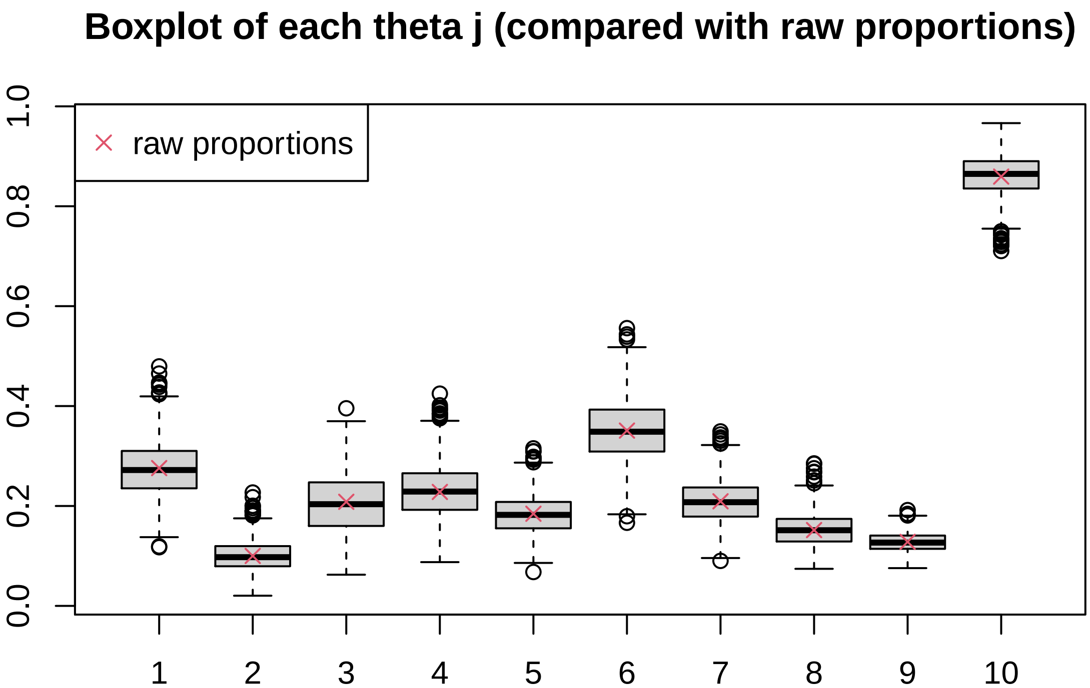
```
  temp_log_U <- log(runif(1))
  if (temp_log_U <= log_ratio) {
    res[i+1, 2] <- temp_beta
  } else {
    res[i+1, 2] <- res[i, 2]
  }
}
res <- tail(res, -20000)
```

```
xgrid <- seq(-4, 3, .01)
ygrid <- seq(-14, 1, .05)
z <- matrix(nrow=length(xgrid), ncol=length(ygrid))
for (i in 1:length(xgrid)) {
  for (j in 1:length(ygrid)) {
    z[i, j] <- exp(log_posterior(xgrid[i], ygrid[j]))
  }
}
par(mfrow=c(1,2))
contour(x=xgrid, y=ygrid, z=z, drawlabels=F,
        main='Contour plot for posterior density',
         xlab='log(alpha/beta)', ylab='log(alpha+beta)')
plt_df <- res[sample(nrow(res), 1e3), ]
plot(plt_df[, 1], plt_df[, 2],
     xlim=c(-4, 3), ylim=c(-14, 1), pch=16, cex=.5,
     main='1000 draws from the posterior',
     xlab='log(alpha/beta)', ylab='log(alpha+beta)')
```




```
b <- exp(plt_df[, 2]) / (exp(plt_df[, 1])+1)
a <- exp(plt_df[, 2]) - b
```

```
res <- matrix(0, length(a), length(n))
for (l in 1:length(a)) {
  for (j in 1:length(n)) {
    res[l, j] <- rbeta(1, a[l]+y[j], b[l]+n[j]-y[j])
  }
}

boxplot.matrix(res, main="Boxplot of each theta j (compared with raw proportions)")
points(factor(1:10), y/n, col=2, pch=4)
legend("topleft", col=2, pch=4, legend="raw proportions")
```



From above figure we see that the posterior median (black bar) is approximately the same as raw proportion (red cross).

(d) & (e) & (f)

The 95% posterior interval for the average underlying proportion of traffic that is bicycles is as follows:

```
quantile(rowMeans(res), c(.05, .95))
```

```
##        5%       95%
## 0.2468407 0.2940282
```

The 95% posterior interval for the number of those vehicles that are bicycles is as follows:

```
floor(100*quantile(rowMeans(res), c(.05, .95)))
```

```
##  5% 95%
##  24  29
```

In application I think this is a good reference to consider but we might need to measure multiple times and get more data points to make the inference more reliable.

I think it's reasonable to assume θ_j are beta random variables, at least it's convenient to use beta-binomial setting.

Problem 3 (Exercise 5.17)

- Noninformative: $p(\mu, \sigma) \propto \sigma^{-1}$.
- Subjective: $\mu \sim \mathrm{U}(-0.1, 0.3)$ and $\sigma \sim \mathrm{U}(0.1, 0.2)$. I make θ_j concentrates around a meaningful range while still allowing it to fluctuate for different basketball players.
- Weakly informative: $\mu \sim \mathrm{U}(-0.1, 1)$ and $\sigma \sim \mathrm{U}(0.1, 2)$. This only tells us that we don't expect the training improvement to be too negative.