R code and output for Problem 3.12

I am assuming the log-linear model throughout: $E(y_i \mid x_i, \alpha, \beta) = e^{\alpha + \beta x_i}$. The first step is to enter the data and center the x_i variables (for numerical stability).

```
y0 <- c(734, 516, 754, 877, 814, 362, 764, 809, 223, 1066)
y <- c(24, 25, 31, 31, 22, 21, 26, 20, 16, 22)
x0 <- 1976:1985
x <- x0 - mean(x0)
```

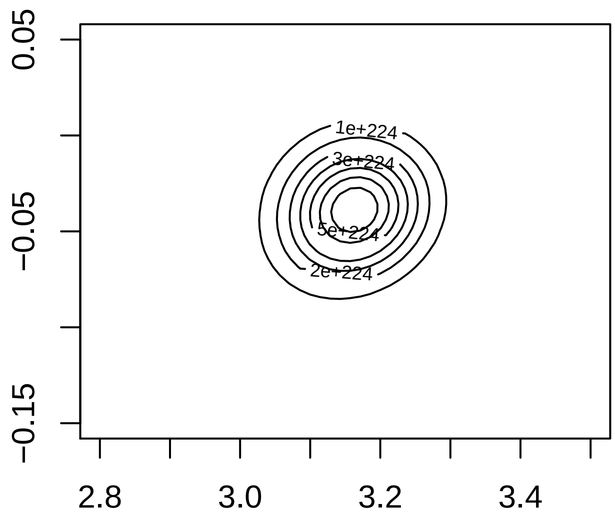
Next, run regression on $\log(y)$ to get a rough idea of the final answer. (We could also do a generalized linear model fit.)

```
summary(mod1 <- lm(log(y)~x))

##
## Call:
## lm(formula = log(y) ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23766 -0.08553 -0.06471  0.15553  0.22154
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.15179    0.05337  59.054 7.51e-12 ***
## x           -0.04044    0.01858  -2.176   0.0612 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1688 on 8 degrees of freedom
## Multiple R-squared:  0.3719,Adjusted R-squared:  0.2934
## F-statistic: 4.737 on 1 and 8 DF,  p-value: 0.06121
```

Now get a contour plot of the posterior (up to normalizing constant. Note that I arrived at the limits for the x- and y-axes for the contour plot by trial and error. I am using flat (constant) priors for both α and β.

```
al <- seq(2.8, 3.5, , 50)
bet <- seq(-.15, .05, , 50)

na <- length(al)
nb <- length(bet)
z <- matrix(0, na, nb)

for (i in 1:na){
```

```
  for (j in 1:nb){
                z[i, j] <- exp( al[i]*sum(y) + bet[j]*sum(x*y) - sum(exp(al[i] + bet[j]*x)))
        }
}

contour(al, bet, z)
```



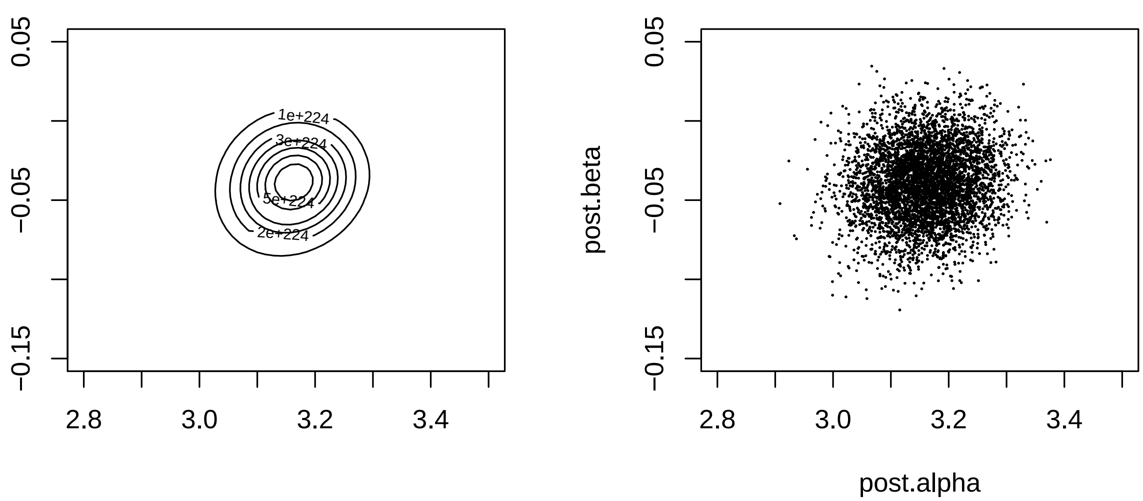
The contours look very nearly normal and independent.

Here I compute the posterior mode using the function `optim()`. Note that `optim()` minimizes its argument, so the function `logp()` I created supplies the negative log likelihood.

```
logp <- function(bb){
  a <- bb[1]
        b <- bb[2]
        -(a*sum(y) + b*sum(x*y) - sum(exp(a + b*x)))
        }

optim(c(3.15, -.04), logp)$par

## [1]  3.1634137 -0.0388174
```

bf 3.12(f)   As in the text, we'll sample from the discretized posterior on a fine grid. random noise is added to the output to make the draws from the posterior look continuous.

```
# marginal distribution for alpha: renormalize
pa <- apply(z, 1, sum)         # sum across the rows to get marginal pdf
pa <- pa/sum(pa)               # normalize so sum is 1.

post.alpha <- post.beta <- rep(NA, 5000)
for (k in 1:5000){
  i <- sample(1:na, 1, prob=pa)
# conditional distribution of beta given alpha
        pb.a <- z[i, ]/sum(z[i, ])
        j <- sample(1:nb, 1, prob=pb.a)
# add random jitter as in BDA3 so sample looks continuous
        post.alpha[k] <- al[i] + runif(1, -.007, .007)
        post.beta[k] <- bet[j] + runif(1, -.002, .002)
}
par(mfrow=c(1, 2))
contour(al, bet, z)
plot(post.alpha, post.beta, xlim=c(2.8, 3.5), ylim=c(-.15, .05), cex=.1)
```

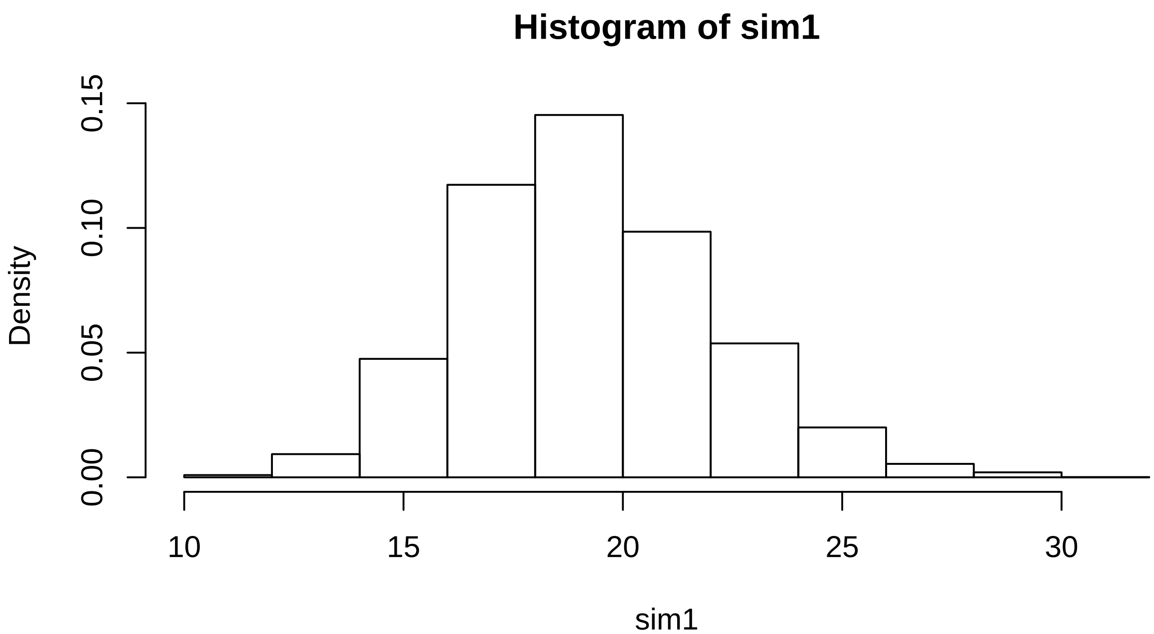


bf 3.12(f)   Posterior distribution of expected value for $x_p = 1986$ and 95% credible interval.

```
xp <- 1986 - mean(x0)
sim1 <- exp(post.alpha + post.beta*xp)
quantile(sim1, c(.025, .975))
```

```
##      2.5%    97.5%
## 14.24817 25.23528
```

```r
hist(sim1, freq=F)
```



bf 3.12(f) Posterior distribution of expected value for $x_p = 1986$ and 95% credible interval.

```r
xp <- 1986 - mean(x0)
sim2 <- rep(NA, length(post.alpha))
for (i in 1:length(sim2)) sim2[i] <- rpois(1, exp(post.alpha[i] + post.beta[i]*xp))
hist(sim2, freq=F)
quantile(sim2, c(.025, .975))
```

```
##  2.5% 97.5%
##    10    31
```


Histogram of sim2
Density
0.00
0.02
0.04
0.06
0.08
10
20
30
40
sim2