Bayesian data analysis demo 3.6

Aki Vehtari, Markus Paasiniemi

2021-09-04

Binomial regression and grid sampling for Bioassay data (BDA3 p. 74-)

ggplot2, and gridExtra are used for plotting, tidyr for manipulating data frames

```
library(ggplot2)
theme_set(theme_minimal())
library(gridExtra)
library(tidyr)
library(dplyr)
library(purrr)
```

The Dataset

```
df1 <- data.frame(
  x = c(-0.86, -0.30, -0.05, 0.73),
  n = c(5, 5, 5, 5),
  y = c(0, 1, 3, 5)
)
```

Trying out *logistic regression*

We employ Maximum Likelihood Estimation in order to find the α and β. Specifically, we employ the logistic regression algorithm.

```
theta <-  df1$y / df1$n
theta
```

```
## [1] 0.0 0.2 0.6 1.0
```

```
df1$theta <- theta
df1
```

```
##       x n y theta
## 1 -0.86 5 0   0.0
## 2 -0.30 5 1   0.2
## 3 -0.05 5 3   0.6
## 4  0.73 5 5   1.0
```

```
log_fits <- glm( theta  ~ x, family = binomial, data=df1 )
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(log_fits)
```

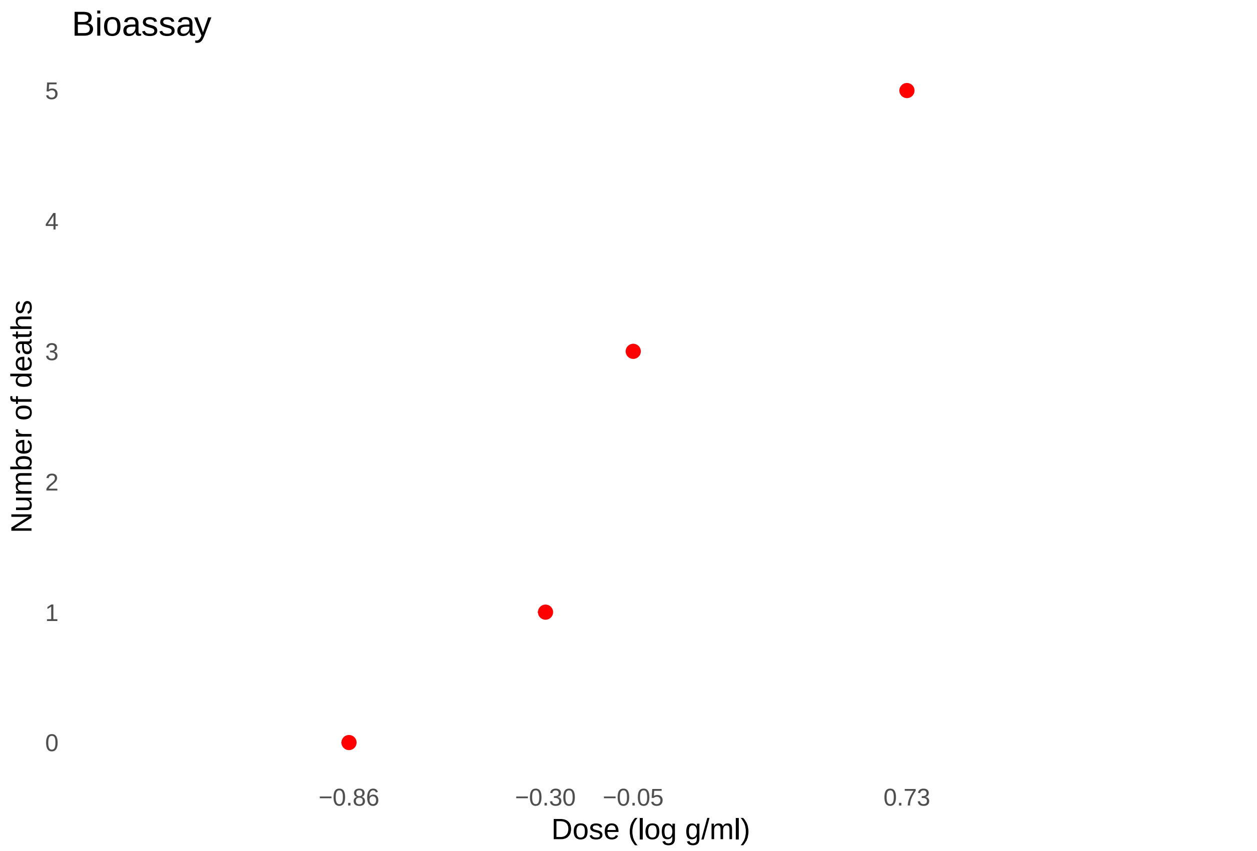
```
##
## Call:
## glm(formula = theta ~ x, family = binomial, data = df1)
```

```
##
## Deviance Residuals:
##        1         2         3         4
## -0.07708   0.03637  -0.02625   0.05473
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8466     2.2787   0.372    0.710
## x             7.7488    10.8957   0.711    0.477
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.158282  on 3  degrees of freedom
## Residual deviance: 0.010948  on 2  degrees of freedom
## AIC: 5.3992
##
## Number of Fisher Scoring iterations: 7
```

As we can see that the maximum likelihood estimate of $(\hat{\alpha}, \hat{\beta})$ is $(0.8, 7.7)$.

Plot data

```
ggplot(df1, aes(x=x, y=y)) +
    geom_point(size=2, color='red') +
    scale_x_continuous(breaks = df1$x, minor_breaks=NULL, limits = c(-1.5, 1.5)) +
    scale_y_continuous(breaks = 0:5, minor_breaks=NULL) +
    labs(title = 'Bioassay', x = 'Dose (log g/ml)', y = 'Number of deaths') +
    theme(panel.grid.major = element_blank())
```



Compute the posterior density in grid

- usually should be computed in logarithms!
- with alternative prior, check that range and spacing of A and B are sensible

```
A = seq(-4, 8, length.out = 50)
B = seq(-10, 40, length.out = 50)
```

Make vectors that contain all pairwise combinations of A and B

```
cA <- rep(A, each = length(B))
cB <- rep(B, length(A))
```

Make a helper function to calculate the log likelihood given a dataframe with x, y, and n and evaluation points a and b. For the likelihood see BDA3 p. 75 `log1p(x)` computes log(x+1) in numerically more stable way.

```
logl <- function(df, a, b)
  df['y']*(a + b*df['x']) - df['n']*log1p(exp(a + b*df['x']))
```

Calculate likelihoods: apply logl function for each observation ie. each row of data frame of x, n and y

```
p <- apply(df1, 1, logl, cA, cB) %>%
  # sum the log likelihoods of observations
  # and exponentiate to get the joint likelihood
  rowSums() %>% exp()
```

Sample from the grid (with replacement)

```
nsamp <- 1000
samp_indices <- sample(length(p), size = nsamp,
                       replace = T, prob = p/sum(p))
samp_A <- cA[samp_indices[1:nsamp]]
samp_B <- cB[samp_indices[1:nsamp]]
```

Add random jitter, see BDA3 p. 76

```
samp_A <- samp_A + runif(nsamp, (A[1] - A[2])/2, (A[2] - A[1])/2)
samp_B <- samp_B + runif(nsamp, (B[1] - B[2])/2, (B[2] - B[1])/2)
```

Create data frame

```
samps <- data_frame(ind = 1:nsamp, alpha = samp_A, beta = samp_B) %>%
  mutate(ld50 = - alpha/beta)
```

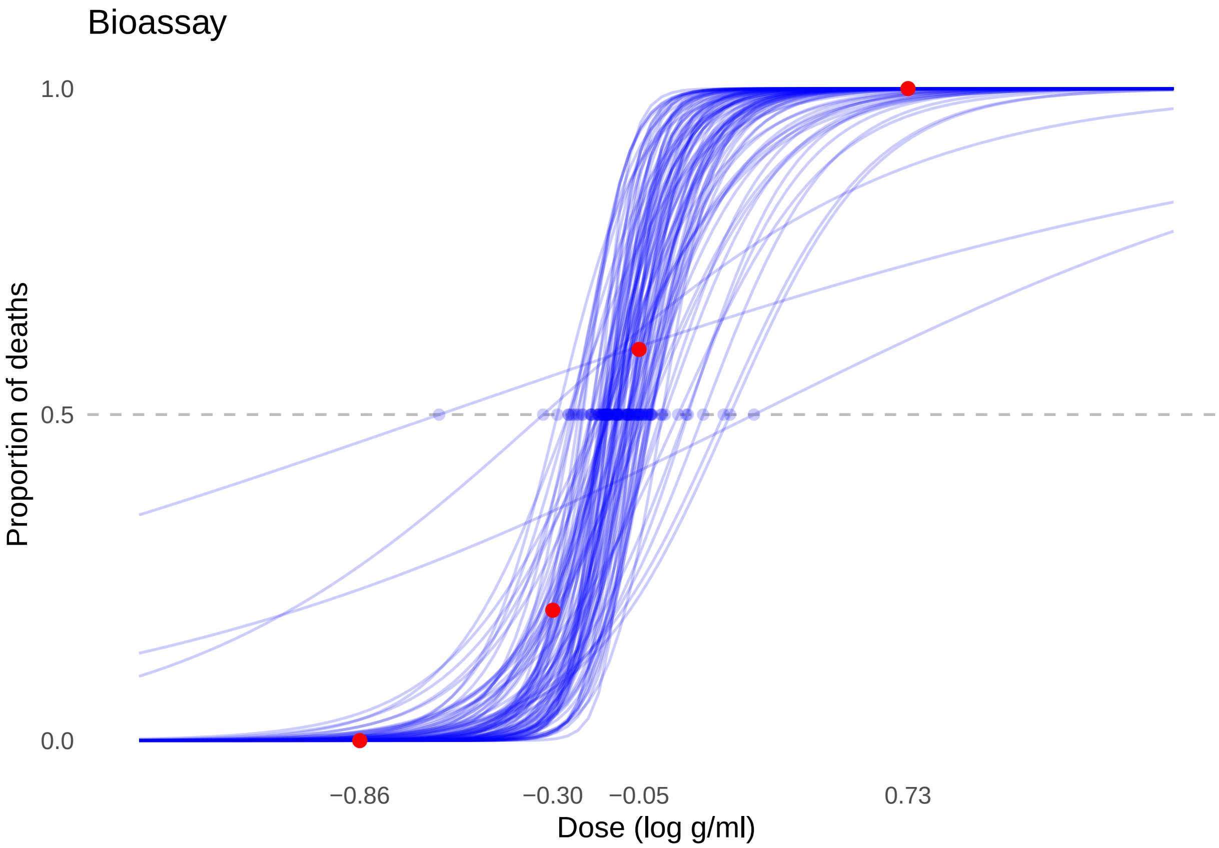
```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
```

Plot draws of logistic curves

```
invlogit <- plogis
xr <- seq(-1.5, 1.5, length.out = 100)
dff <- pmap_df(samps[1:100,], ~ data_frame(x = xr, id=..1,
                                           f = invlogit(..2 + ..3*x)))
ppost <- ggplot(df1, aes(x=x, y=y/n)) +
  geom_line(data=dff, aes(x=x, y=f, group=id), linetype=1, color='blue', alpha=0.2) +
  geom_point(size=2, color='red') +
  scale_x_continuous(breaks = df1$x, minor_breaks=NULL, limits = c(-1.5, 1.5)) +
  scale_y_continuous(breaks = seq(0,1,length.out=3), minor_breaks=NULL) +
  labs(title = 'Bioassay', x = 'Dose (log g/ml)', y = 'Proportion of deaths') +
  theme(panel.grid.major = element_blank())
```

add 50% deaths line and LD50 dots

```
ppost + geom_hline(yintercept = 0.5, linetype = 'dashed', color = 'gray') +
  geom_point(data=samps[1:100,], aes(x=ld50, y=0.5), color='blue', alpha=0.2)
```



Create a plot of the posterior density

```
# limits for the plots
xl <- c(-2, 8)
yl <- c(-2, 40)
pos <- ggplot(data = data.frame(cA ,cB, p), aes(cA, cB)) +
  geom_raster(aes(fill = p, alpha = p), interpolate = T) +
  geom_contour(aes(z = p), colour = 'black', size = 0.2) +
  coord_cartesian(xlim = xl, ylim = yl) +
  labs(title = 'Posterior density evaluated in grid', x = 'alpha', y = 'beta') +
  scale_fill_gradient(low = 'yellow', high = 'red', guide = F) +
  scale_alpha(range = c(0, 1), guide = F)
```

Plot of the samples

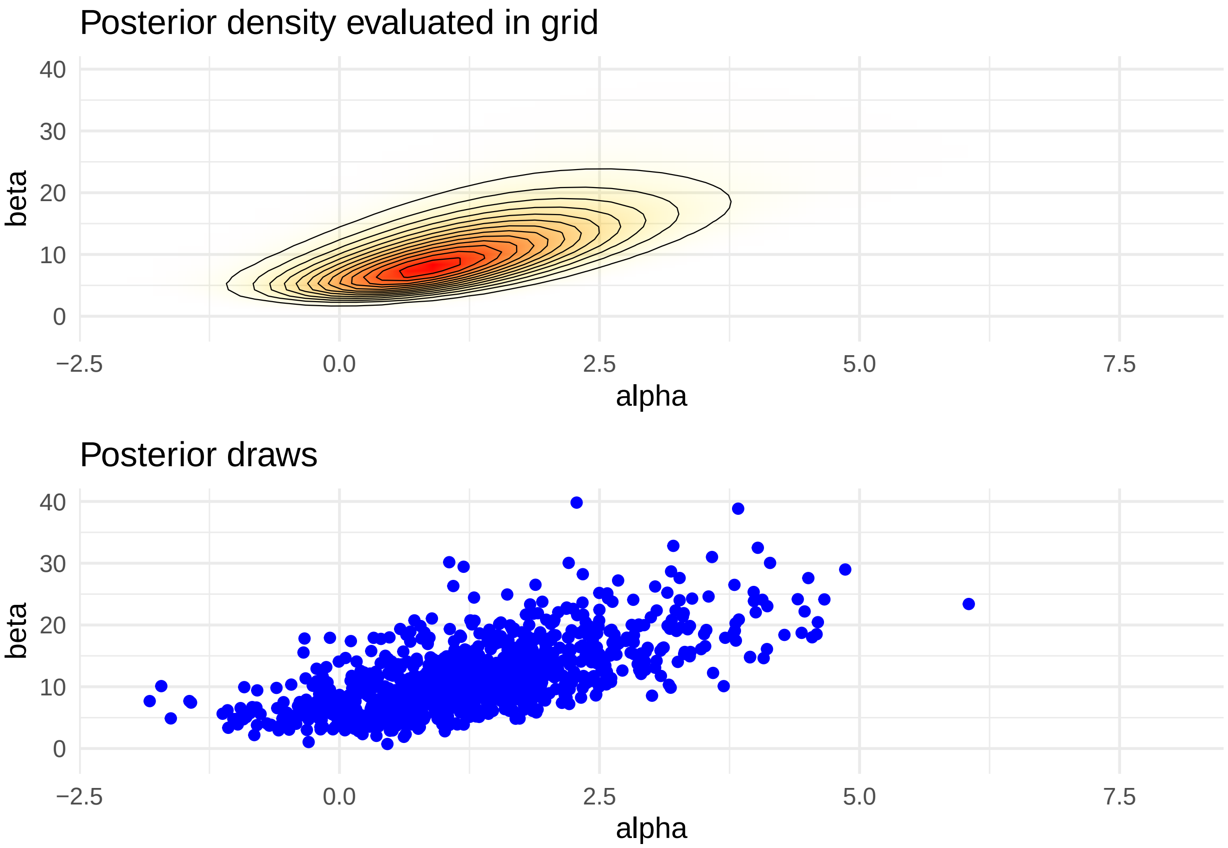
```
sam <- ggplot(data = samps) +
  geom_point(aes(alpha, beta), color = 'blue') +
  coord_cartesian(xlim = xl, ylim = yl) +
  labs(title = 'Posterior draws', x = 'alpha', y = 'beta')
```

Combine the plots

```
grid.arrange(pos, sam, nrow=2)
```

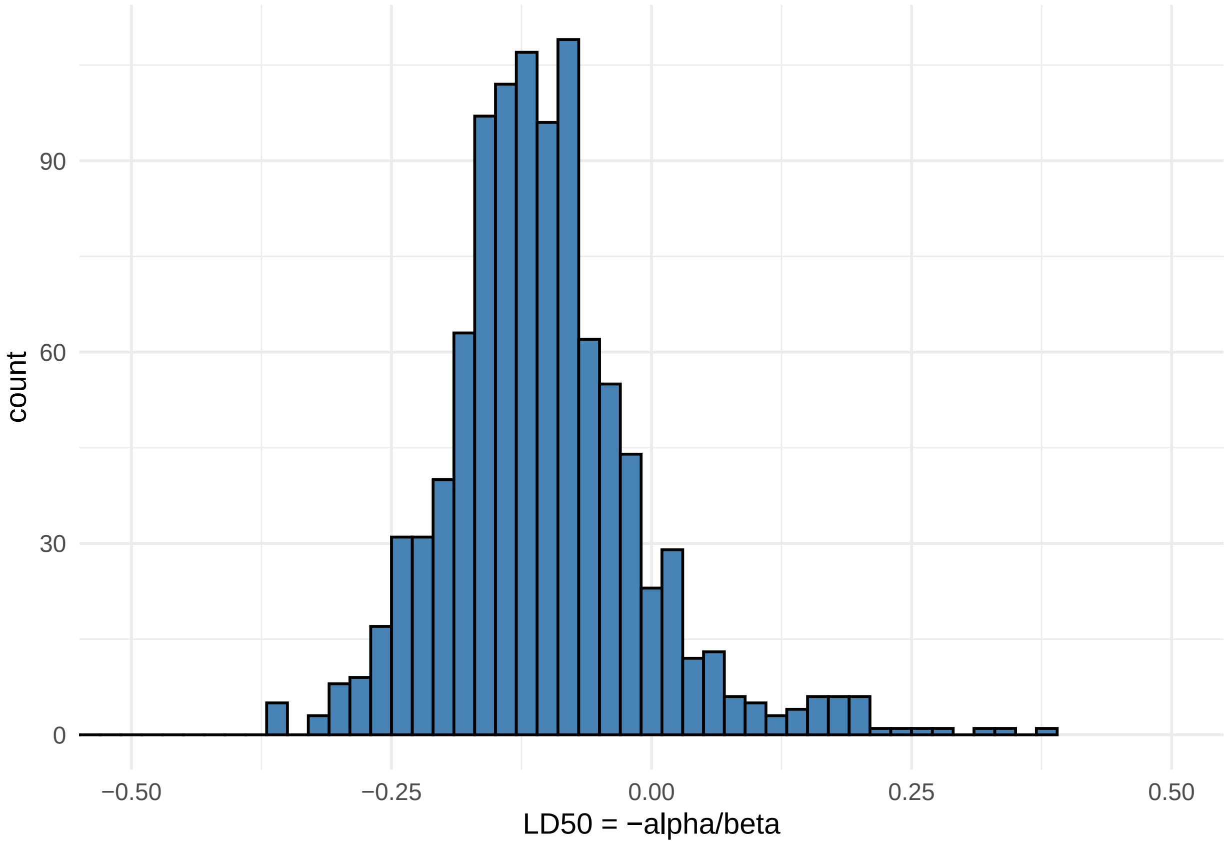
```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



Plot of the histogram of LD50

```
his <- ggplot(data = samps) +
  geom_histogram(aes(ld50), binwidth = 0.02,
                 fill = 'steelblue', color = 'black') +
  coord_cartesian(xlim = c(-0.5, 0.5)) +
  labs(x = 'LD50 = -alpha/beta')
his
```


count
90
60
30
0
−0.50
−0.25
0.00
0.25
0.50
LD50 = −alpha/beta