Tutorial on Bayesian Statistics. Homework from BDA3

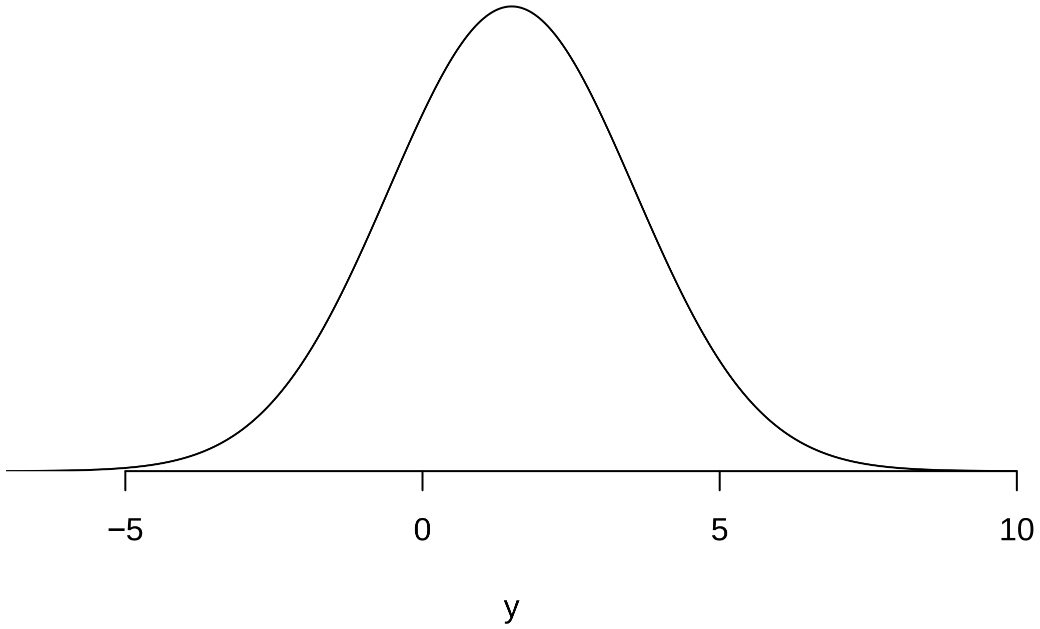
Fernando Hoces de la Guardia

Note: Some of the solutions presented here are have been reverse engineered from here.

- 1.1
- 1a:

$$
\begin{aligned}
p(y) &= \frac{1}{2}\left(p(y|\theta=1)+p(y|\theta=2)\right) \\
&= \frac{1}{2}\left(N(y|1,2^2)+N(y|2,2^2)\right)
\end{aligned}
$$

```
domain          <- seq(-7,10,.02)
dens            <- 0.5*dnorm(domain,1,2) + 0.5*dnorm(domain,2,2)
plot (domain, dens, ylim=c(0,1.1*max(dens)),
type="l", xlab="y", ylab="", xaxs="i",
yaxs="i", yaxt="n", bty="n", cex=2)
```



- 1b:

$$
\begin{aligned}
p(\theta=1|y=1) &= \frac{p(\theta=1)p(y=1|\theta=1)}{\sum_{i=1}^{2} p(\theta=i)p(y=1|\theta=i)} \\
&= \frac{0.5N(1|1,4)}{\sum_{i=1}^{2} 0.5N(1|i,4)}
\end{aligned}
$$

```
p.theta.1       <- function(sigma) {
  res1          <- (0.5*dnorm(1,1,sigma)) /(sum(0.5*dnorm(1,c(1,2),sigma)))
  return(res1)
  }
```

Evaluating the last expression in the respective cumulative distribution function we get:0.5312. **Note: even though we are adding "discrete" number of probabilities, we are still in the continuous space (but for $y = 1$) and should evaluate the probabilities in the density function.**

- 1c:

Table 1: Posterior probabilty of $\theta = 1$, af a function of σ

σ	$p(\theta = 1 \vert y = 1)$
0.25	0.9997
0.5	0.8808
1	0.6225
2	0.5312
4	0.5078
8	0.502

- 1.7 *Let's Make a Deal*
  Calculate the probability of winning for each box after one of the empty boxes has been revealed and is not a winning box.

Lets define the following events:
* A : The participant chose the right box at the beginning.
* B : The host opens a particular box, among the unchosen ones, such that is empty.
* C : Among the unchosen boxes the host chooses a empty box.

And let's compute the probabilities of each of this events.

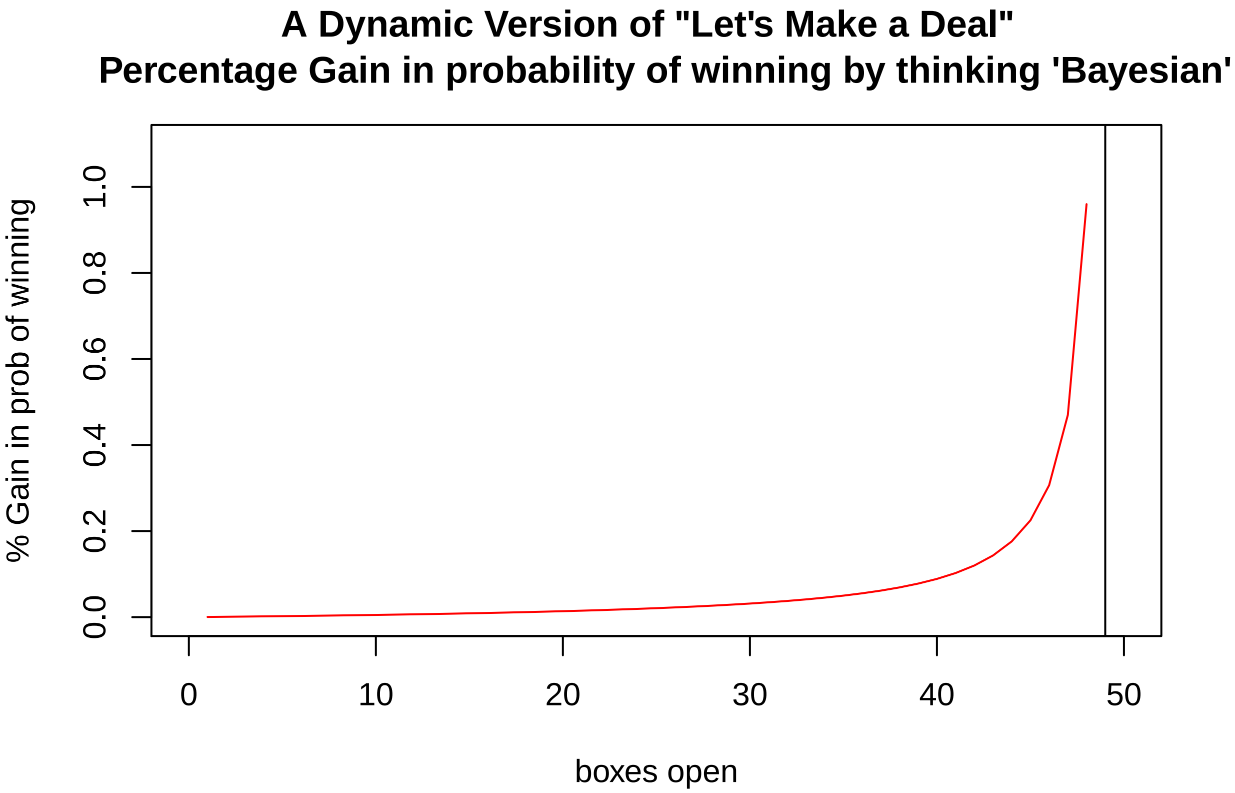
$$
\begin{aligned}
Pr(A) &= 1/3 \\
Pr(C) &= 1/2 \\
Pr(B) &= Pr(B|A)Pr(A) + Pr(B|\neg A)Pr(\neg A) = (1/2) * (1/3) + Pr(B|\neg A) * (2/3) \\
&= 1/6 + 2/3 * (Pr(B|\neg A, C)Pr(C) + Pr(B|\neg A, \neg C)Pr(\neg C)) \\
&= 1/6 + 2/3 * (1 * (1/2) + 0 * (1/2)) = 1/2
\end{aligned}
$$

Using Bayes' theorem we have that the probability of choosing the right box from the beginning, conditional on a unchosen box being revealed as a losing one is:

$$Pr(A|B) = \frac{Pr(A)Pr(B|A)}{Pr(B)} = \frac{(1/3) * (1/2)}{1/2} = \frac{1}{3}$$

The participant's chances are not equal across remaining boxes! She is worst of staying with her original choice (33% probability of wining instead of 50%!).

More generally if there were n boxes in total and i boxes where revealed, we have that the wrong way of updating the probabilities ($1/(n - i)$) and the Bayesian update ($\frac{i+n*(n-1-i)}{n*(n-i)*(n-i-1)}$) differ significantly as $i \to n$. For example the following graph plots both probabilities of winning in a contest with 50 boxes as the host opens i boxes.



Looking at the graph it seems that the advantages of thinking in a Bayesian fashion are certainly parameter-specific. Also notice that the player here chooses a "stubborn" strategy, I suspect that if she changes boxes in a optimal way the improvement in her chances will be slightly less. Maybe that is the reason why we don't think in a Bayesian fashion all the time.

- 2.1

$$\begin{aligned} P(\theta) &= Beta(4,4) \\ P(y|\theta) &= Bin(y|n,\theta) \\ \Rightarrow P(\theta|y) &= Beta(4+y, 4+(n-y)) \end{aligned}$$

The **wrong** way to answer the question would be:

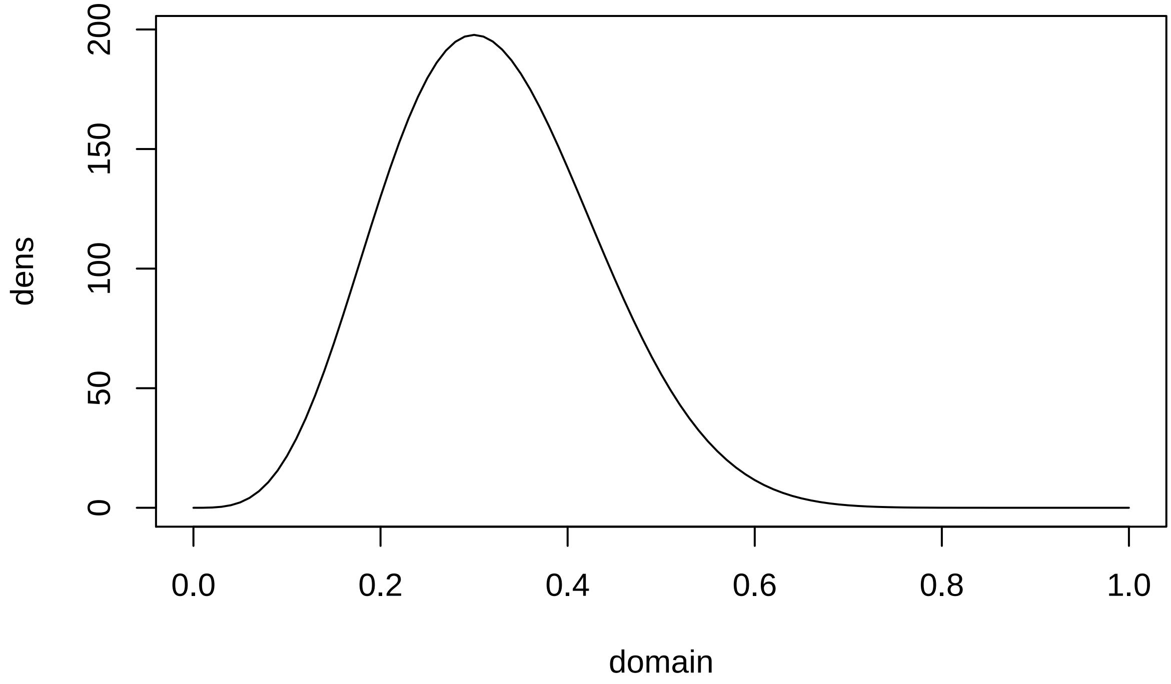
$$P(\theta|y<3) \propto \sum_{i=0}^{2} Beta(4+i, 4+(n-i))$$

The **right** way to answer the question would be:

$$\begin{aligned} P(y<3|\theta) &= \sum_{i=0}^{2} Bin(i|n,\theta) \\ \Rightarrow P(\theta|y) &\propto \sum_{i=0}^{2} \binom{n}{i} Beta(4+i, 4+(n-i)) \end{aligned}$$

In this case some part of the proportionality constant *does* matter.

```
domain <- seq(0,1,.01)
dens = apply(sapply(0:2,function(x) choose(10,x)*dbeta(domain,4+x,4+10-x)),1,sum)
plot(domain, dens, type="l")
```



- 2.14
- 2.14a Deriving the posterior for a normal likelihood with known variance, unknown mean, and using a normal prior. Slide 15 here

Note: a good reminder of the main conjugacy relationships can be found here

- 5.3 Reproducing results of section 5.5

```
#Data:
school.id       <- LETTERS[1:8]
effect          <- c(28,8 ,-3,7 ,-1,1 ,18,12)
se.effect       <- c(15,10,16,11,9 ,11,10,18)

pool.est        <- sum(effect*se.effect^-2)/sum(se.effect^-2)
pool.var        <- sum(se.effect^-2)^-1
pool.ci         <- c(-1.96,1.96)*pool.var^.5 + pool.est
```

The pooled estimated effect and variance are 7.69 and 16.58, with a 95% CI of [-0.3, 15.67].

Posterior simulation under ther hierarchical model
Using the identity:

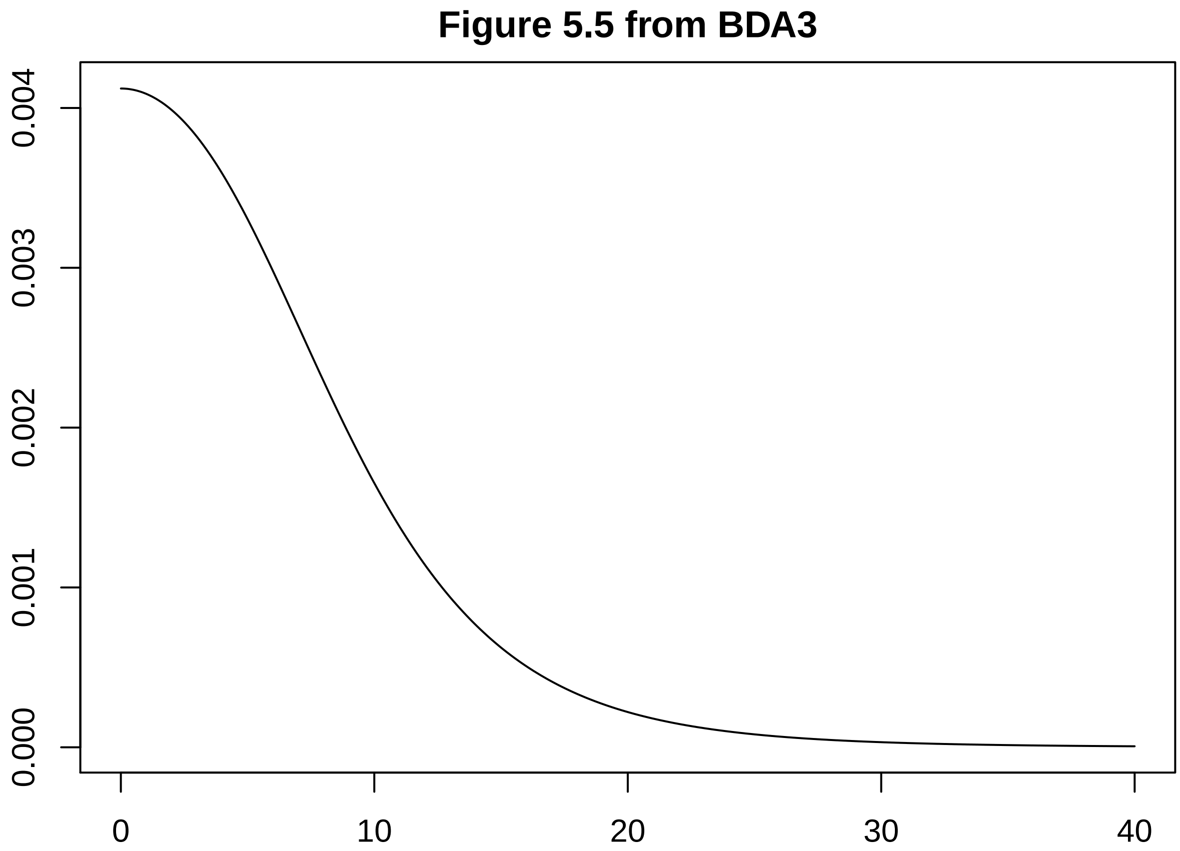
$$p(\theta, \mu, \tau|y) = p(\tau|y)p(\mu|\tau, y)p(\theta|\mu, \tau, y)$$

And the results from BDA in equation 5.17, 5.20 and 5.21, we code the joint posterior:

```r
# Eqn 5.17 of BDA3: theta| mu, tau, y ~ N(post.theta.j, post.v.theta.j). Where:
post.theta.j    <- function(mu,tau,j)
{
    ( effect[j] / (se.effect[j]^2) + mu / (tau^2) ) /
    ( 1 / ( se.effect[j]^2 ) + 1 / ( tau^2 ) )
}

post.v.theta.j  <- function(tau,j)
{
    1 / (1 / ( se.effect[j]^2 ) + 1 / ( tau^2 ) )
}

# Eqn 5.20 of BDA3: mu| tau, y ~ N(post.mu.hat, post.v.mu). Where:
post.mu.hat     <- function(tau)
{
    sum( effect * 1 / ( se.effect^2 + tau^2 ) ) /
    sum( 1 / ( se.effect^2 +tau^2 ) )
}

post.v.mu       <- function(tau)
{
    ( sum( 1 / ( se.effect^2 +tau^2 ) ) )^-1
}

# Eqn 5.21 of BDA3: p(tau | y)
marginal.tau    <- function(tau)
{
    hyper.prior(tau) * ( post.v.mu(tau)^(1/2) ) *
    prod(
      ( ( se.effect^2 + tau^2 )^(-1/2) ) * exp(
      - ( (effect - post.mu.hat(tau) )^2 ) /
        ( 2 * ( se.effect^2 + tau^2 ) )
      )
      )
}

# Testing alternative: this function is not been used currently. But I'm curoius as to why I don't
# get the same results as with "marginal.tau".
marginal.tau1   <- function(tau) {
    marg.post   <- 1
        for (i in 1:length(effect)) {
      marg.post   <- ((se.effect[i]^2 + tau^2)^(-1/2)) *
                 exp(-((effect[i] - post.mu.hat(tau))^2) / (2 * (se.effect[i]^2 + tau^2))) *
                 marg.post
        return(hyper.prior(tau) * (post.v.mu(tau)^.5) * marg.post)
    }
}
```

Define a hyper-prior and draw 1000 samples from each distribution (for all 8 schools).

```r
set.seed(142857)
samps           <- 1000

hyper.prior     <-  function(tau) 1
tau.grid        <-  seq(0.001 , 40 , length=samps)
pdf.tau         <-  sapply( tau.grid , function(x) marginal.tau(x) )
pdf.tau         <-  pdf.tau / sum( pdf.tau )

s.tau           <- sample(tau.grid,samps,prob=pdf.tau, replace=TRUE)
s.mu            <- sapply(s.tau,function(x) rnorm(1,post.mu.hat(x),(post.v.mu(x))^0.5))
s.theta         <- NULL
for (j in 1:length(school.id)) {
  s.theta[[j]]         <- sapply(1:samps,
                            function(x)
                            rnorm(1,
                                  post.theta.j(s.mu[x],s.tau[x],j),
                                  (post.v.theta.j(s.tau[x],j))^0.5
                                  ) )
  }
par(mfrow=c(1,1))
par(mar = rep(2, 4))
plot(tau.grid,pdf.tau, type="l", main="Figure 5.5 from BDA3", xlab=expression(tau), ylab="Density")
```



The sampling method in BDA3 suggest to apply the inverse method from the posterior of τ. I don't do this for two reasons: (i) I'm not sure the posterior has a closed for solution for its inverse, and (ii) given that I already have the density, I can directly draw from that distribution sampling using the `sample` command (which leads me to think that this command applies the inverse method, but).

```
# Store the simulations in a #samps x 8 matrix
s.theta         <- matrix(unlist(s.theta), ncol = 8, byrow = FALSE)

# Obtain quantiles for each school
s.theta.sort    <- apply(s.theta, 2, sort)
p               <- t( apply(s.theta.sort, 2, function(x) quantile(x,c(.025,.25,.5, .75, .975),type=1)) )
p               <- round(p,3)
```

Table 5.3 from BDA3:

School					
	2.5%	25%	median	75%	97.5%
A	-1.517	6.275	10.086	15.274	33.036
B	-4.568	3.624	7.682	11.594	20.251
C	-10.038	1.954	6.699	10.697	19.491
D	-5.786	3.827	7.539	11.633	20.098
E	-8.444	1.178	5.309	9.039	16.515
F	-8.304	2.325	6.373	10.336	18.247
G	-0.422	6.026	10.192	14.357	25.733
H	-6.794	4.223	8.296	12.752	25.081

Here we reproduce figure 5.8

```
par(mfrow=c(1,2))
domain           <- c(-20,60)
hist(s.theta[,1],
     breaks=10,
     xlab="Effect in School A",
     main="",
     xlim=domain)
hist(apply(s.theta,1,max),
     breaks=10,
     xlab="Largest Effect",
     main="",
     xlim=domain)
title(main="Figure 5.8 from BDA3")
```

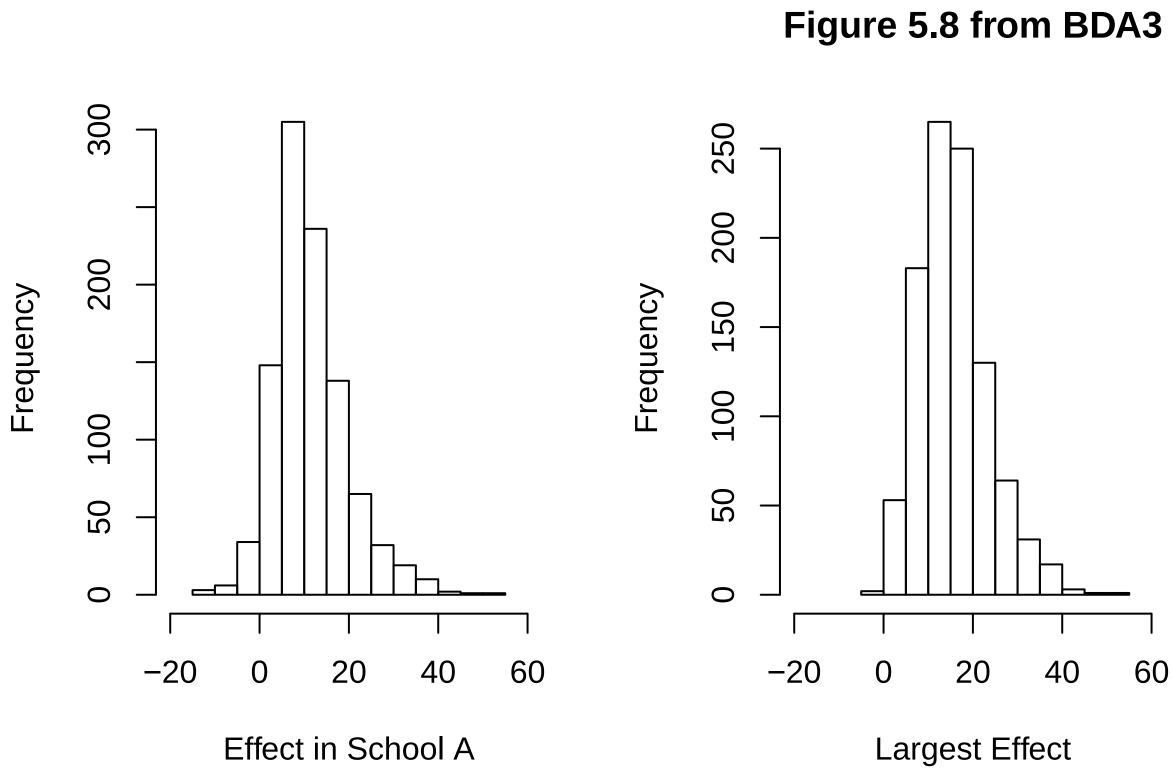


Figure 5.8 from BDA3

This last figure ("largest effect") is a good example of one the main advantage of a fully Bayesian hierarchical model: once we have correctly simulated the posterior, we can test all kinds of complicated hypothesis.

- 5.3a (i) - For each school j, the probability that its coaching program is the best of eight: (**Important:** do not sort each posterior).

```r
aux1              <- apply(s.theta,1,max)
best              <- apply(1*(s.theta==aux1), 2,mean)
```

Table 2: Probability that each coaching program is the best among the eight schools

School	Probability of having the best coaching program
A	0.231
B	0.113
C	0.094
D	0.093
E	0.065
F	0.068
G	0.21
H	0.126

- 5.3a (ii) - For each school j, the probability that its coaching program is better than other school k:

```
p               <- sapply( 1:8,
                          function(y) sapply( 1:8,
                                             function(x)
                                               mean( 1 * ( s.theta[,x] > s.theta[,y] ) )
                                             )
                          )
```

Table 3: Probability that school j (row) has a better program that school k (column)

School j/School k	A	B	C	D	E	F	G	H
A	0	0.632	0.672	0.626	0.744	0.707	0.5	0.613
B	0.368	0	0.578	0.512	0.611	0.555	0.377	0.468
C	0.328	0.422	0	0.443	0.557	0.505	0.318	0.421
D	0.374	0.488	0.557	0	0.622	0.574	0.368	0.467
E	0.256	0.389	0.443	0.378	0	0.445	0.268	0.346
F	0.293	0.445	0.495	0.426	0.555	0	0.32	0.417
G	0.5	0.623	0.682	0.632	0.732	0.68	0	0.59
H	0.387	0.532	0.579	0.533	0.654	0.583	0.41	0

- 5.3b (i) - Now with $\tau = \infty$ compute for each school j, the probability that it has the best coaching program:
  With $\tau = \infty$ each school posterior effect is independent $\theta_j \sim N(y_y, \sigma_j^2)$. The probability of a school having the best coaching program is:
  Wrong way to do it:

$$
\begin{aligned}
p(\theta_j > max_{i \neq j}\{\theta_i\}) &= \prod_{i \neq j} p(\theta_j > \theta_i) \\
&= \prod_{i \neq j} \Phi(\frac{\theta_j - \theta_i}{\sigma_i})
\end{aligned}
$$

Right way to do it:

$$
\begin{aligned}
p(\theta_j > max_{i \neq j}\{\theta_i\}) &= \int \prod_{i \neq j} p(\theta_j > \theta_i) \phi(\theta_j | y_j, \sigma_j) d\theta_j \\
&= \int \prod_{i \neq j} \Phi\left(\frac{\theta_j - \theta_i}{\sigma_i}\right) \phi(\theta_j | y_j, \sigma_j) d\theta_j
\end{aligned}
$$

This integral has to be solved numerically:

```
set.seed(142857)
best            <-  sapply(1:8,
                      function(y) mean( sapply( 1:1000 ,
                        function(x)
                          prod( pnorm( (
                            rnorm( 1 , effect[y] , se.effect[y] ) - effect[-y] ) /
                              se.effect[-y] ) ) )
                           )
                           )
# Ad-hoc normalization:
best            <- best/sum(best)
```

Table 4: Probability that each coaching program is the best among the eight schools (with $\tau = \infty$)

School	Probability of having the best coaching program
A	0.5599
B	0.033
C	0.0265
D	0.0364
E	0.0034
F	0.0136
G	0.1615
H	0.1656

- 5.3b (ii) - Now with $\tau = \infty$ compute for each school j, the probability that its coaching program is the better than other school k:

$$\begin{aligned} p(\theta_i > \theta_j) &= p\left(-\frac{y_j - y_i}{\sqrt{\sigma_i^2 + \sigma_j^2}} > \frac{(\theta_j - \theta_i) - (y_j - y_i)}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) \\ &= \Phi\left(\frac{y_i - y_j}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) \end{aligned}$$

The following table presents the different values for the expression above:

```
p               <- sapply(1:8,function(x)
                   sapply(1:8,function(y)
                  pnorm( q = 0, mean = (effect[x] - effect[y]) / sqrt(se.effect[x]^2 + se.effect[y]^2)
                  sd = 1 )
                  ) )
# Force all elementens in the diagonal to zero.
p               <- p - .5 * diag(8)
```

Table 5: Probability that j (row) has a better program that school k (column). With $\tau = \infty$

School j/School k	A	B	C	D	E	F	G	H
A	0	0.8664	0.9212	0.8705	0.9513	0.9267	0.7105	0.7527
B	0.1336	0	0.7201	0.5268	0.7482	0.6811	0.2398	0.423
C	0.0788	0.2799	0	0.3033	0.4566	0.4184	0.1329	0.2667
D	0.1295	0.4732	0.6967	0	0.7132	0.6501	0.2297	0.4063
E	0.0487	0.2518	0.5434	0.2868	0	0.444	0.0789	0.2591
F	0.0733	0.3189	0.5816	0.3499	0.556	0	0.1264	0.301
G	0.2895	0.7602	0.8671	0.7703	0.9211	0.8736	0	0.6146
H	0.2473	0.577	0.7333	0.5937	0.7409	0.699	0.3854	0

- 5.3c The estimated differences between the closed form solutions (5.3b) and the bayesian analysis (5.3a) is that the latter presents less extreme probability estimates (shrinkage)
- 5.3d If $\tau = 0$, then all effects are the same so the probabilities can be 0 or 1 for all schools (all are the largest effect and the smallest at the same time)

- **5.13 - Bicycles**

```
#Load data
y                <- c(16, 9  , 10 , 13 , 19 , 20 , 18 , 17 , 35 , 55 )
n                <- c(74, 99 , 58 , 70 , 122, 77 , 104, 129, 308, 119)
```

- 5.13a $y_i \sim Bin(\theta_i, n_i)$ where n_i represents the *total* number of vehicles (bicycles + other vehicles). $\theta_i \sim Beta(\alpha, \beta)$ the prior distribution of biking rates for each street. We set a noninformative hyperprior $p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}$. This implies that the **joint posterior** distribution is the the following (same as in equation 5.6 in BDA3):

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

- 5.13b Compute the marginal posterior of θ, conditional on α, β. For the beta-binomial case we have that given the hyper-parameters, each θ_j has a posterior distribution $Beta(\alpha + y_j, \beta + n_j - y_j)$. Assuming exchangeability:

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

Now we compute the posterior marginal of (α, β). Given that we do have a closed form solution in step 2, we compute the ratio of (\ref{bic.joint.post1}) and (\ref{bic.cond.post.theta1}).

$$
p(\alpha, \beta|y) \propto \prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)}
$$

Centering our grid around the methods of moments estimates for (α_0, β_0):

$$
\begin{aligned}
\hat{\mu} &= 0.1961 = \frac{\hat{\alpha_0}}{\hat{\alpha_0} + \hat{\beta_0}} \\
\hat{\sigma^2} &= 0.0111 = \frac{\hat{\alpha_0}\hat{\beta_0}}{(\hat{\alpha_0} + \hat{\beta_0})^2(\hat{\alpha_0} + \hat{\beta_0} + 1)}
\end{aligned}
$$

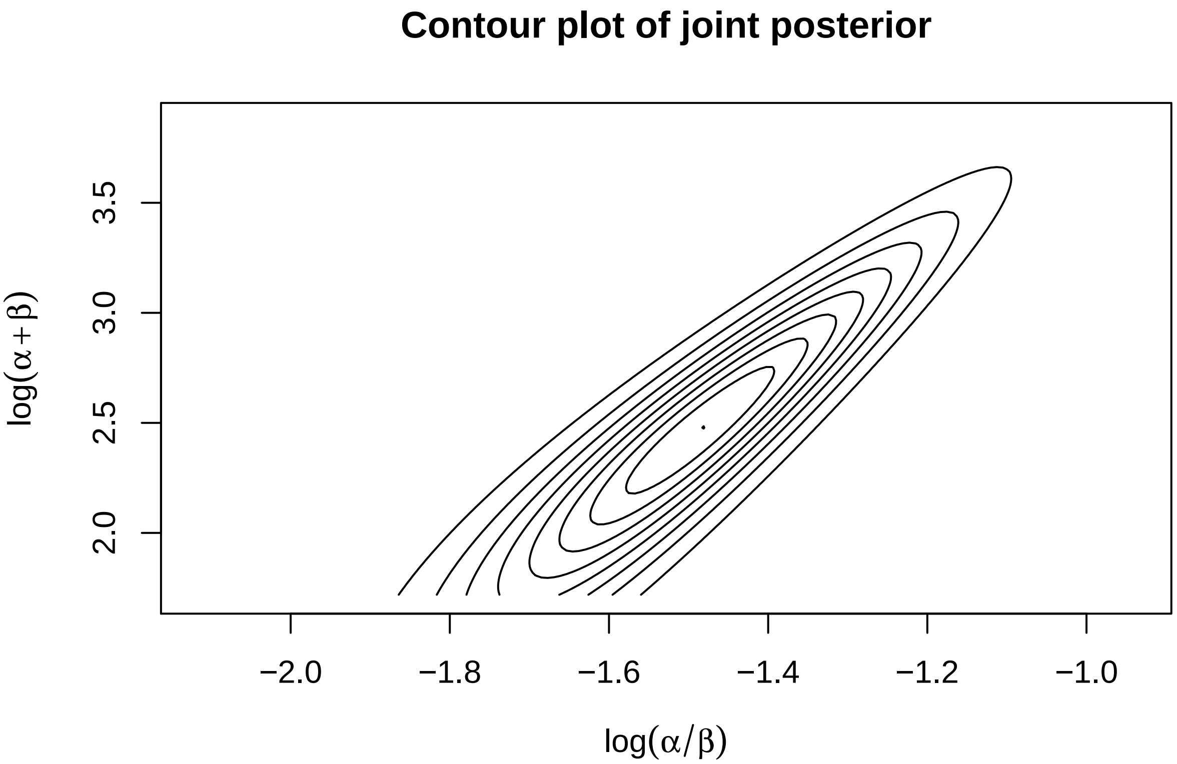
Solving form $(\hat{\alpha_0}, \hat{\beta_0})$:

```
#Here 'x' represents alpha and beta
dslnex          <- function(x) {
    z           <- numeric(2)
    z[1]        <- x[1]/(x[1]+x[2]) - mean(y/n)
    z[2]        <- x[1]*x[2]/(((x[1]+x[2])^2)*(x[1]+x[2]+1)) - sd(y/n)^2
    z
}

sol1            <- nleqslv(c(1,1), dslnex)
res1            <- paste("(",round(sol1$x[1],1), ",", round(sol1$x[2],1), ")",sep="")
```

We get: $(\hat{\alpha_0}, \hat{\beta_0}) = (2.6, 10.6)$.

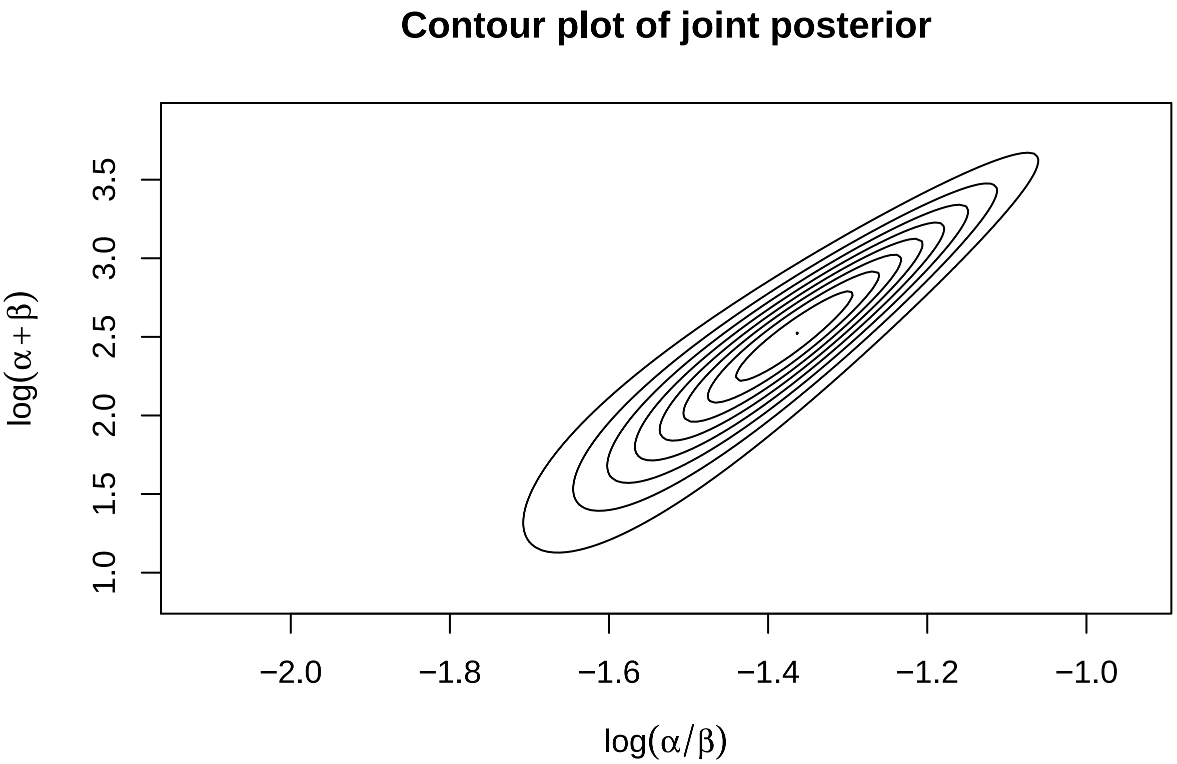
We center the grid (approximately) around that initial estimate and expand the grid to cover up to a factor of 4 of each parameter. The result is plotted in the following figure:

```
bic.marg.post.phi <-   function(alpha, beta) {
  post          <-  1
  #notice the censoring in n (the gamma(.) function in R cannot handle large values)
  for (i in 1:length(y)) {
    if (n[i] > 100) n[i] = 100
    post  = post * (
      ( ( gamma(alpha + beta) ) /
        ( gamma(alpha) * gamma(beta) ) ) *
      ( ( gamma(alpha + y[i] ) * gamma(beta + n[i] - y[i]) ) /
        ( gamma(alpha + beta + n[i]) ) )
      )
  }
  # The hyper prior is defined below
  bic.hyper.prior(alpha,beta) * post
}

bic.hyper.prior <-  function(alpha,beta)
{
    alpha*beta*(alpha + beta)^(-5/2)
}

v1              <-  seq(log(sol1$x[1]/sol1$x[2])*1.5,log(sol1$x[1]/sol1$x[2])/1.5,length.out =151)
v2              <-  seq(log(sol1$x[1]+sol1$x[2])/1.5,log(sol1$x[1]+sol1$x[2])*1.5,length.out =151)
beta            <-  exp(v2)/(exp(v1)+1)
alpha           <-  exp(v2+v1)/(exp(v1)+1)

post.dens       <-  outer(alpha,beta,function(x1,x2) log(bic.marg.post.phi(x1, x2)) )
post.dens       <-  exp(post.dens - max(post.dens))
post.dens       <-  post.dens/sum(post.dens)

contours        <-  seq(min(post.dens), max(post.dens) , length=10)
contour(v1, v2, post.dens,
        levels=contours,
        xlab=expression( log(alpha/beta) ),
        ylab=expression( log(alpha+beta) ),
        xlim=c( min( v1 ), max( v1 ) ) ,
        ylim=c( min( v2 ), max( v2 ) ),
        drawlabels=FALSE,
        main="Contour plot of joint posterior")
```



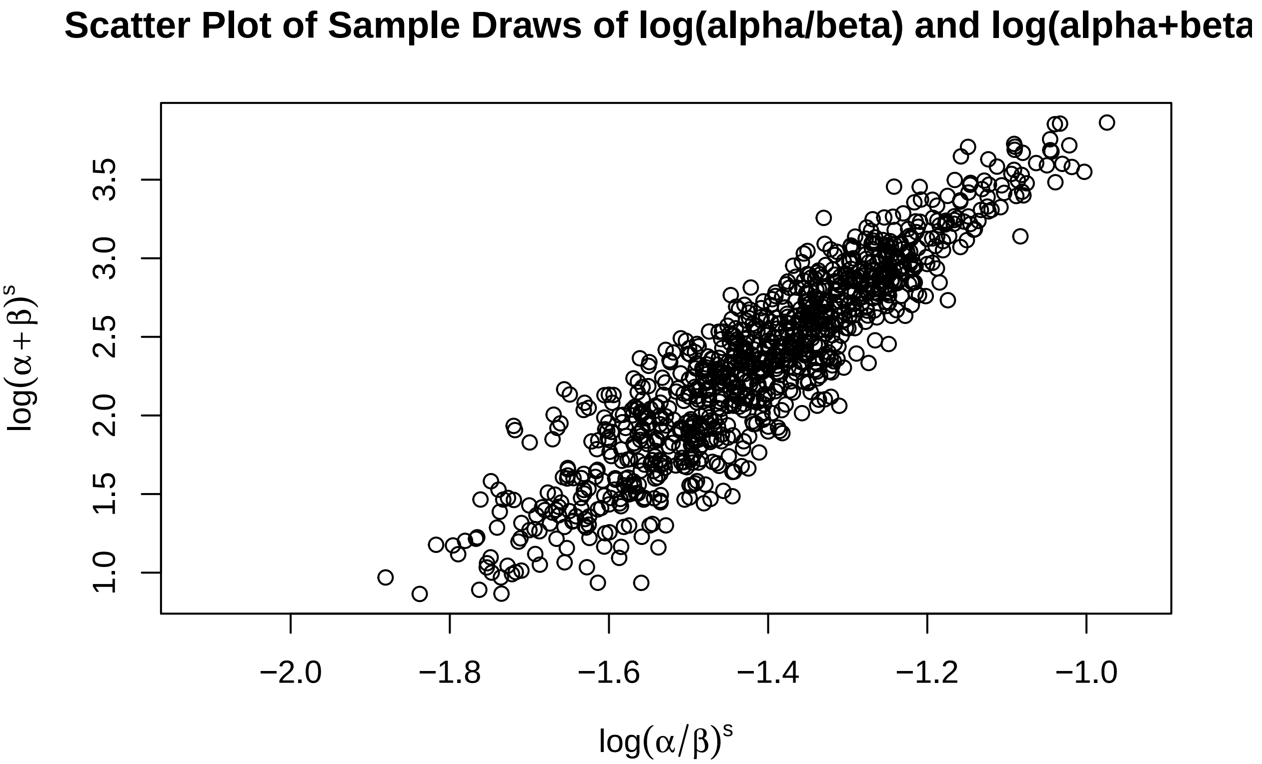
Adjust the grid and repeat:

```
v1              <-  seq(log(sol1$x[1]/sol1$x[2])*1.5,log(sol1$x[1]/sol1$x[2])/1.5,length.out =151)
v2              <-  seq(log(sol1$x[1]+sol1$x[2])/3,log(sol1$x[1]+sol1$x[2])*1.5,length.out =151)
beta            <-  exp(v2)/(exp(v1)+1)
alpha           <-  exp(v2+v1)/(exp(v1)+1)

post.dens       <-  outer(alpha,beta,function(x1,x2) log(bic.marg.post.phi(x1, x2)) )
post.dens       <-  exp(post.dens - max(post.dens))
post.dens       <-  post.dens/sum(post.dens)

contours        <-  seq(min(post.dens), max(post.dens) , length=10)
contour(v1, v2, post.dens,
        levels=contours,
        xlab=expression( log(alpha/beta) ),
        ylab=expression( log(alpha+beta) ),
        xlim=c( min( v1 ), max( v1 ) ) ,
        ylim=c( min( v2 ), max( v2 ) ),
        drawlabels=FALSE,
        main="Contour plot of joint posterior")
```



Draw samples (α^s, β^s) from $p(\alpha, \beta|y)$ (finally!). Here we repeat the procedure used in section 3.(v) of the book replication document.

```
samps       = 1000
v1.dens     = apply(post.dens ,1, sum)
s.v1        = sample(v1,samps, replace=TRUE, prob = v1.dens)

#Select the colum of the joint density corresponding to a specific value of v1 (p(v2|v1))
cond.v2     = function(x)
{
  post.dens[which(v1 == s.v1[x]),]
}
#Sample a value of v2 according the the conditional probatility above
s.v2        = sapply(1:samps,function(x) sample(v2,1,replace=TRUE,prob=cond.v2(x)))

#Add a uniform random jitter centered at zero with with equal to the grid spacing. This will make the s
grid.v1     = v1[2]-v1[1]
grid.v2     = v2[2]-v2[1]
s.v2        = s.v2 + runif(length(s.v2),-grid.v2/2,grid.v2/2)
s.v1        = s.v1 + runif(length(s.v1),-grid.v1/2,grid.v1/2)
plot(s.v1, s.v2, xlab=expression(log(alpha/beta)^s), ylab=expression(log(alpha+beta)^s), xlim=c(min(v1)
```

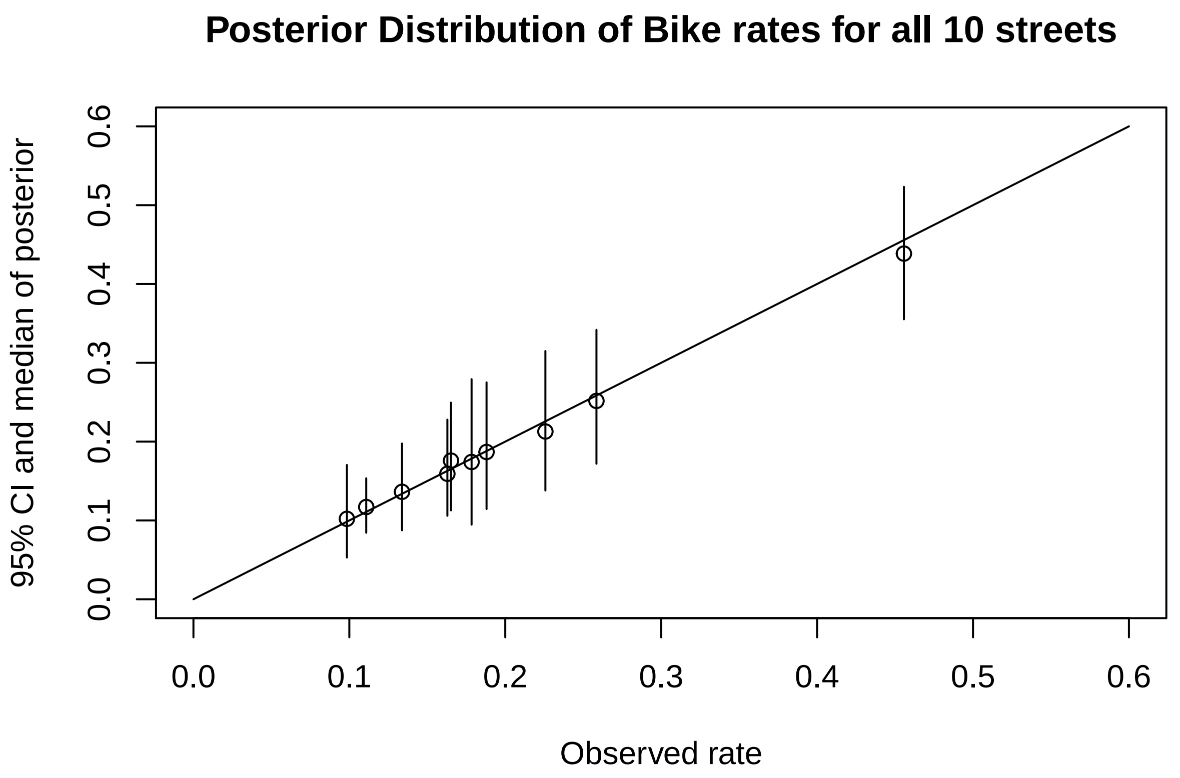


By applying the inverse of the transformation we recover the marginal distribution of the original hyperparameters.

```
s.beta       = exp(s.v2)/(exp(s.v1)+1)
s.alpha      = exp(s.v2+s.v1)/(exp(s.v1)+1)
```

- 5.13c For each draw of ϕ^s, draw a sample of θ from $p(\theta|\phi^s, y)$

```
s.beta       = exp(s.v2)/(exp(s.v1)+1)
s.alpha      = exp(s.v2+s.v1)/(exp(s.v1)+1)
theta.dist   = sapply(1:10, function(x) rbeta(1000,s.alpha+y[x], s.beta + n[x] - y[x]))
theta.dist   = apply(theta.dist,2,sort)
plot(0:600/1000,0:600/1000, type="l", xlab="Observed rate",ylab="95% CI and median of posterior")
jitter.x     = y/n + runif(length(y),-0.01,0.01)
points(jitter.x,theta.dist[500,])
segments(jitter.x,theta.dist[25,], jitter.x,theta.dist[975,] )
title(main="Posterior Distribution of Bike rates for all 10 streets")
```



The estimated proportions are almost the same as the raw proportions (no shrinkage).

- 5.13d We generate 1000 draws from a $Beta(\alpha^s, \beta^s)$ where the parameters come from the draws obtained above:

```
s.theta         <- rbeta(1000, shape1 =s.alpha , shape2 = s.beta)
CI.num          <- round(s.theta[order(s.theta)][c(25,975)],2)
CI.str          <- paste("(" , CI.num[1] , "," , CI.num[2] , ")")
```

The posterior interval for $\hat{\theta} = (0.01, 0.48)$

- 5.13e If a new street is opening with 100 vehicles per day. The posterior interval predicts with 95% confidence that between 1 and 48. This CI is not so informative as it covers almost all the possible oberved bike rates.
- 5.13f The beta assumption might not have been so reasonable as the posterior estimates did not show much shrinkage.
- **5.14**

- *5.14a* Set up a model in which the total number of vehicles observed at each location j follows a Poisson distribution with parameter θ_j, the 'true' rate of traffic per hour at the location. Assign a gamma population distribution for the parameters θ_j and a noninformative hyperprior distribution. Write down the joint posterior distribution.

Now we have that $n_j \sim Poi(\lambda = \theta_j)$ and $\theta_j \sim Gamma(\alpha)$. And the joint posterior is:

$$
\begin{aligned}
p(\theta, \alpha, \beta|y) &\propto p(\alpha, \beta) \times p(\theta|\alpha, \beta) \times p(y|\theta, \alpha, \beta) \\
p(\theta, \alpha, \beta|y) &\propto 1 \times \prod_{j=1}^{10} Gamma(\theta_j|\alpha, \beta) \times \prod_{j=1}^{10} Poisson(y_j|\theta_j) \\
&= \prod_{j=1}^{10} \frac{\beta^\alpha}{\Gamma(\alpha)} \theta_j^{\alpha-1} exp(-\beta\theta) \times \frac{\theta_j^{y_i} exp(-\theta_j)}{!y_j} \\
&\propto \frac{\beta^{n\alpha}}{\Gamma(\alpha)^n} exp(-\sum \theta_j(1+\beta)) \prod_{j=1}^{10} \theta_j^{\alpha+y_j-1}
\end{aligned}
$$

- *5.14b*
  Then compute the marginal posterior of θ, conditional on α, β. For the gamma-poisson case we have that given the hyper-parameters, each θ_j has a posterior distribution $Gamma(\alpha + n_j, \beta + 1)$. Assuming exchangeability:

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

Now we compute the posterior marginal of (α, β). Given that we do have a closed form solution in step 2, we compute the ratio of (\ref{bic.joint.post2}) and (\ref{bic.cond.post.theta2}).

$$
p(\alpha, \beta|y) \propto \frac{\beta^{n\alpha}}{\Gamma(\alpha)^n} \prod_{i=1}^{n} \frac{\Gamma(\alpha + y_i)}{(\beta+1)^{\alpha+y_i}}
$$

Centering our grid around the methods of moments estimates for (α_0, β_0):

$$
\begin{aligned}
\hat{\mu} &= 116 = \frac{\hat{\alpha_0}}{\hat{\beta_0}} \\
\hat{\sigma^2} &= 5141.7778 = \frac{\hat{\alpha_0}}{\hat{\beta_0}^2}
\end{aligned}
$$

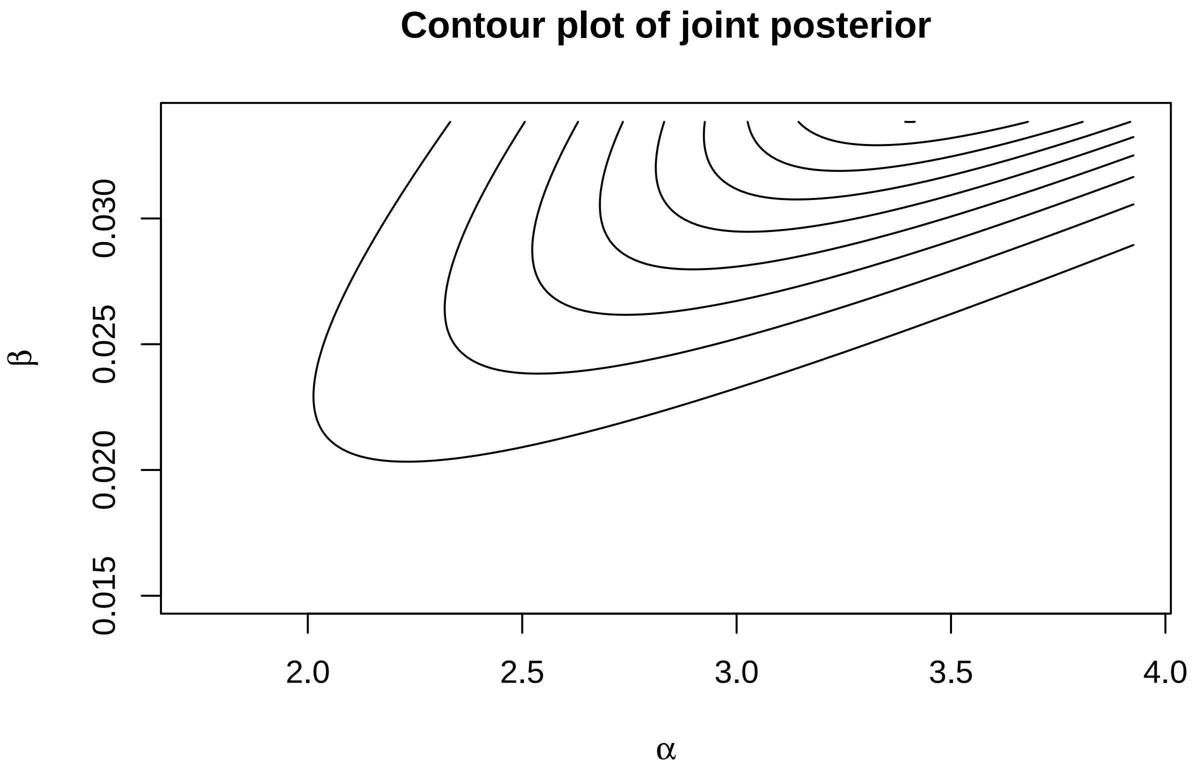
Solving for $(\hat{\alpha_0}, \hat{\beta_0})$:

```
#Here 'x' represents alpha and beta
dslnex          <- function(x) {
    z           <- numeric(2)
    z[1]        <- x[1]/(x[2]) - mean(n)
    z[2]        <- x[1]/(x[2]^2) - sd(n)^2
    z
}

sol1            <- nleqslv(c(1,1), dslnex)
res1            <- paste("(",round(sol1$x[1],1), ",", round(sol1$x[2],2), ")",sep="")
```

We get: $(\hat{\alpha_0}, \hat{\beta_0}) = (2.6, 0.02)$.

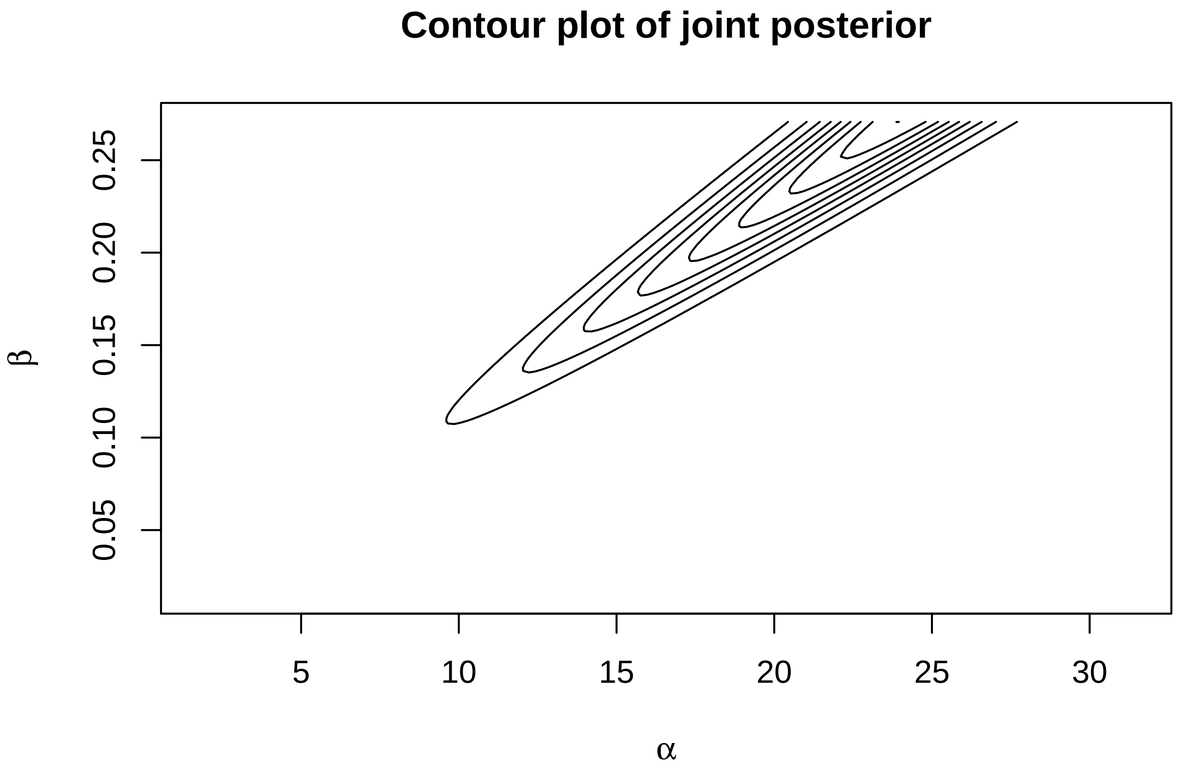
We center the grid (approximately) around that initial estimate and expand the grid to cover up to a factor of 4 of each parameter. The result is plotted in the following figure:

```
bic.marg.post.phi <-   function(alpha, beta) {
  log.post          <-  0
  #notice the censoring in n
  for (i in 1:length(n))
  {
    if (n[i] > 100) n[i]  <-  100
    log.post        <-  log.post + log(gamma( alpha+n[i] )) - (alpha+n[i])*log((beta + 1))
  }
  # The hyper prior is defined below
  log(bic.hyper.prior2(alpha,beta)) + log.post + (length(n)*alpha)*log(beta) - length(n)*log(gamma(alph
}

bic.hyper.prior2 <-  function(alpha,beta) {
  1
}


alpha           <-  seq(sol1$x[1]/1.5,sol1$x[1]*1.5,length.out =151)
beta            <-  seq(sol1$x[2]/1.5,sol1$x[2]*1.5,length.out =151)

post.dens       <-  outer(alpha,beta,function(x1,x2) bic.marg.post.phi(x1, x2) )
post.dens       <-  exp(post.dens - max(post.dens))
post.dens       <-  post.dens/sum(post.dens)



contours        <- seq(min(post.dens), max(post.dens), length=10)
contour(alpha, beta, post.dens,levels=contours, xlab=expression(alpha), ylab=expression(beta), xlim=c(mi
```



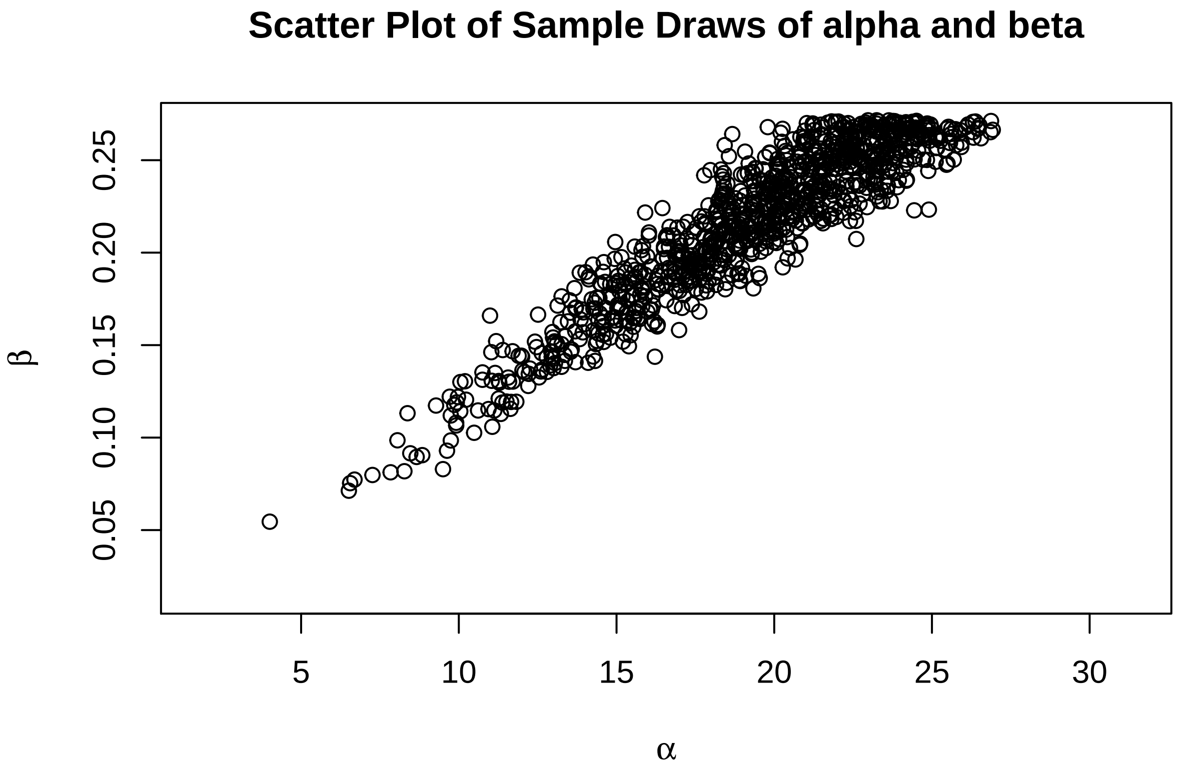
Adjust the grid and repeat:

```
alpha           <-  seq(sol1$x[1]/1.5,sol1$x[1]*12,length.out =151)
beta            <-  seq(sol1$x[2]/1.5,sol1$x[2]*12,length.out =151)

post.dens       <-  outer(alpha,beta,function(x1,x2) bic.marg.post.phi(x1, x2) )
post.dens       <-  exp(post.dens - max(post.dens))
post.dens       <-  post.dens/sum(post.dens)

contours        <- seq(min(post.dens), max(post.dens) , length=10)
contour(alpha, beta, post.dens,levels=contours, xlab=expression(alpha), ylab=expression(beta), xlim=c(mi
```



Draw samples (α^s, β^s) from $p(\alpha, \beta|y)$.

```
samps           <-  1000
alpha.dens      <-  apply(post.dens ,1, sum)
s.alpha         <-  sample(alpha,samps, replace=TRUE, prob = alpha.dens)

#Select the colum of the joint density corresponding to a specific value of v1 (p(beta|alpha))
cond.beta       <-  function(x) {
  post.dens[which(alpha == s.alpha[x]),]
}
#Sample a value of v2 according the the conditional probatility above
s.beta          <-  sapply(1:samps,function(x) sample(beta,1,replace=TRUE,prob=cond.beta(x)))

#Add a uniform random jitter centered at zero with with equal to the grid spacing. This will make the s
grid.alpha      <-  alpha[2]-alpha[1]
grid.beta       <-  beta[2]-beta[1]
s.beta          <-  s.beta + runif(length(s.beta),-grid.beta/2,grid.beta/2)
s.alpha         <-  s.alpha + runif(length(s.alpha),-grid.alpha/2,grid.alpha/2)
plot(s.alpha, s.beta, xlab=expression(alpha), ylab=expression(beta), xlim=c(min(alpha),max(alpha)) , yl
```



Note: regardless of how much I change the range of α and β I dont seem to cover the whole graph.

- *5.14c* Given the previous result we can say that the posterior is not integrable.
- *5.14d* I don't know how to alter it such that it becomes integrable.
- **10.1**

- *10.1a*
  If $\theta \sim N(\mu, \sigma_\theta)$ then R draws $y^{(r)}$ from will have a simulation standard error of $\hat{\sigma_\theta}/\sqrt{R}$. Hence in order to be within $0.1\sigma_\theta$ we need $R = 100$.

- *10.1b*
  For the simulation excercise we choose $\mu = 0, \sigma_\theta = 1$. We perform R draws and for each set of simulated numbers, we compute the 2.5% percentile $\{y^r\}_{p=0.025}$ and its difference with the theoretical percentile (-1.96), we repeat this excercise 100 times and look at the average of the difference for different values of R.

```
y_reps          <- apply(
                     sapply( 1:100,
                       function(x) sapply(10^(1:5),
                         function(x) abs(-1.96 - quantile(rnorm(x, mean=0, sd=1) , c(.025) ) ) )
                           ), 1, mean
                         )
```

Table 6: Standard Error of Simulations

R	$\lvert\{y^r\}_{p=0.025}+1.96\rvert$
10	0.6378
100	0.1822
1000	0.067
10000	0.0233
100000	0.0072

Note: both results don't match.

- **10.4a**
  I follow this proof.

We want to prove that the conditional distribution of θ

We want to prove that $g(\theta|acceptance) = p(\theta)$. The pdf of a drawn from using the rejection sampling algorithm follows:

$$Pr(\theta \leq \theta^*|\theta \text{ is accepted }) = Pr(\theta \leq \theta^*|U \leq \frac{p(\theta)}{Mg(\theta)})$$

$$= \frac{Pr(\theta \leq \theta^* \text{ and } U \leq \frac{p(\theta)}{Mg(\theta)})}{Pr(U \leq \frac{p(\theta)}{Mg(\theta)})}$$

$$= \frac{\int_{-\infty}^{\theta^*}\int_0^{\frac{p(\theta)}{Mg(\theta)}} g(\theta)dud\theta}{\int_{-\infty}^{\infty}\int_0^{\frac{p(\theta)}{Mg(\theta)}} g(\theta)dud\theta}$$

$$= \frac{\int_{-\infty}^{\theta^*}\left[\frac{p(\theta)}{Mg(\theta)} - 0\right] g(\theta)dud\theta}{\int_{-\infty}^{\infty}\left[\frac{p(\theta)}{Mg(\theta)} - 0\right] g(\theta)dud\theta}$$

$$= \int_{-\infty}^{\theta^*} p(\theta)d\theta \quad \blacksquare$$

Note: I was note able to map perfectly this proof with this other one. that seems more complete. For the proof presented here, my main doubt regards the step between the second and third line.

- **11.2**
  (To do)
- **11.3**
  (To do)
- **11.6a**
  (To do)
- **12**
  Install and run examples in STAN [DONE]

- **14**
  (To do)
  Suggestion: Run a logit regression using a frequentist (through ML) and Bayesian (through GS or MH) and compare the results.

- **15**
  (To do)
  Suggestion: Run a full HLM example and compare with a fixed effect regression.