STA 602 HW5

William Tirone

5.1

```r
#reading in data
school1 = read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/school1.dat')
school2 = read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/school2.dat')
school3 = read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/school3.dat')
```

a)

Referencing lecture 8 p.5:

$$p(\theta|x,\sigma^2) \propto exp\{-\frac{(\theta-\mu_n)^2}{2\tau_n^2}\} \sim N(\mu_n,\tau_n^2)$$
$$\mu_n = \frac{\kappa_0}{\kappa_n}\mu_0 + \frac{n}{\kappa_n}\bar{x}$$
$$\tau_n^2 = \sigma^2/\kappa_n$$

and using results from PH p. 76

```r
part_a = function(data) {

  # prior
  mu0 = 5
  s20 = 4
  k0 = 1
  nu0 = 2

  # data
  n = length(data$V1)
```

```r
  data_bar = mean(data$V1)
  s2 = var(data$V1)

  # posterior inference
  kn = k0 + n
  nun = nu0 + n
  mun = (k0 * mu0 + n*data_bar) / kn
  s2n = (nu0 * s20 + (n-1)*s2 + k0*n*(data_bar - mu0)^2/(kn)) / nun

  # sampling
  s2.postsample = 1/rgamma(10000,nun/2,s2n*nun/2)
  theta.postsample = rnorm(10000, mun, sqrt(s2.postsample/kn))

  # CIs
  sdCI = quantile(sqrt(s2.postsample),c(0.025,0.975))
  thetaCI = quantile(theta.postsample,c(0.025,0.975))

  #posterior predictive
  Y_tilde = rnorm(10000,theta.postsample,sqrt(s2.postsample))

  list(mun,thetaCI,s2n,sdCI,s2.postsample,theta.postsample,Y_tilde)

}
```

Posterior Means and Confidence Intervals

```r
s1 = part_a(school1)
cat("Posterior mean, School 1: ", s1[[1]], "\n",
    "School 1 95% CI for Post mean : ", s1[[2]], "\n",
    "School 1 95% CI for Post SD : ", s1[[4]], "\n")
```

```
Posterior mean, School 1:  9.292308
 School 1 95% CI for Post mean :  7.728327 10.78636
 School 1 95% CI for Post SD :  3.010474 5.210434
```

```r
s2 = part_a(school2)
cat("Posterior mean, School 2: ", s2[[1]], "\n",
    "School 2 95% CI for Post mean : ", s2[[2]], "\n",
    "School 2 95% CI for Post SD : ", s2[[4]], "\n")
```

```
Posterior mean, School 2:  6.94875
```

```
 School 2 95% CI for Post mean :  5.169525 8.715262
 School 2 95% CI for Post SD :  3.357281 5.897955
```

```
s3 = part_a(school3)
cat("Posterior mean, School 3: ", s3[[1]], "\n",
    "School 3 95% CI for Post mean : ", s3[[2]], "\n",
    "School 3 95% CI for Post SD : ", s3[[4]])
```

```
Posterior mean, School 3:  7.812381
 School 3 95% CI for Post mean :  6.143524 9.505724
 School 3 95% CI for Post SD :  2.806738 5.120568
```

b)

```
# posterior thetas from different schools
t1 = s1[[6]]
t2 = s2[[6]]
t3 = s3[[6]]

mean((t1<t2) & (t2<t3)) #1 < 2 < 3
```

```
[1] 0.0053
```

```
mean((t1<t3) & (t3<t2)) #1 < 3 < 2
```

```
[1] 0.0032
```

```
mean((t2<t1) & (t1<t3)) #2 < 1 < 3
```

```
[1] 0.0886
```

```
mean((t2<t1) & (t3<t1)) #2 < 3 < 1
```

```
[1] 0.887
```

```r
mean((t3<t1) & (t1<t2)) #3 < 1 < 2
```

```
[1] 0.0159
```

```r
mean((t3<t2) & (t2<t1)) #3 < 2 < 1
```

```
[1] 0.2195
```

c)

From PH p.72 the posterior predictive distribution was derived as

$$\tilde{Y}|\sigma^2, y_1, ..., y_n \sim N(\mu_n, \tau_n^2 + \sigma^2)$$

```r
y1 = s1[[7]]
y2 = s2[[7]]
y3 = s3[[7]]

mean((y1<t2) & (y2<y3)) #1 < 2 < 3
```

```
[1] 0.1507
```

```r
mean((y1<t3) & (y3<y2)) #1 < 3 < 2
```

```
[1] 0.1507
```

```r
mean((y2<t1) & (y1<y3)) #2 < 1 < 3
```

```
[1] 0.2735
```

```r
mean((y2<t1) & (y3<y1)) #2 < 3 < 1
```

```
[1] 0.4203
```

```r
mean((y3<t1) & (y1<y2)) #3 < 1 < 2
```

```
[1] 0.2172
```

```r
mean((y3<t2) & (y2<y1)) #3 < 2 < 1
```

```
[1] 0.2629
```

d)

```r
# Thetas
mean(t1 > t2 & t1>t3)
```

```
[1] 0.887
```

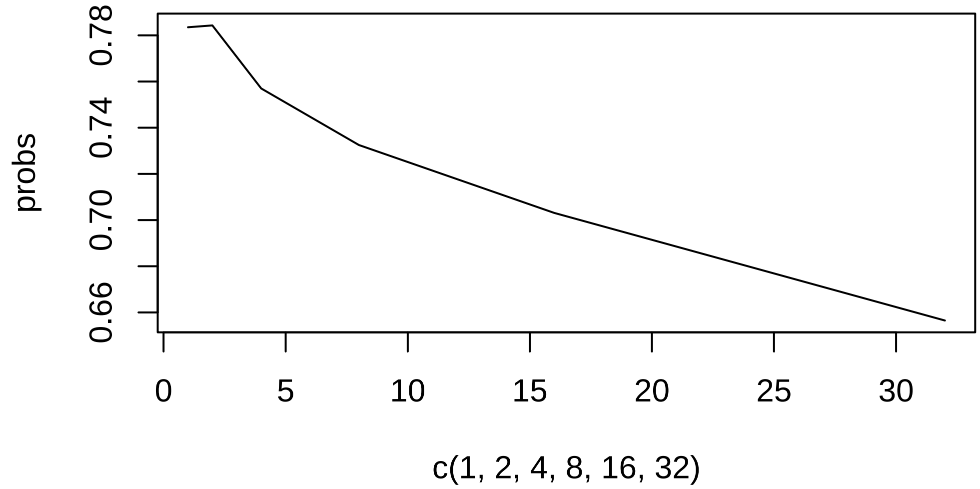
```r
# Post Predict
mean(y1 > y2 & y1>y3)
```

```
[1] 0.4652
```

5.2

```r
n = 16

# A data
ybar.A = 75.2
s.A = 7.3^2

# B data
ybar.B = 77.5
s.B = 8.1^2

# prior
mu0 = 75
sig2.0 = 100
```

```r
# k0, v0 pairs
pairs = list(c(1,1),c(2,2),c(4,4),c(8,8),c(16,16),c(32,32))

# store probabilities here
probs = c()

for (i in 1:6)
  {
  #conditions from pairs
  k0 = pairs[[i]][1]
  nu0 = pairs[[i]][2]

  # posterior inference
  kn = k0 + n
  nun = nu0 + n

  #part A
  mun.A = (k0 * mu0 + n*ybar.A) / kn
  s2n.A = (nu0 * sig2.0 + (n-1)*s.A + (k0*n*(ybar.A - mu0))^2/(kn)) / nun

  # part A sampling
  s2.postsample.A = 1/rgamma(10000,nun/2, s2n.A*nun/2)
  theta.postsample.A = rnorm(10000, mun.A, sqrt(s2.postsample.A/kn))

  #part B
  mun.B = (k0 * mu0 + n*ybar.B) / kn
  s2n.B = (nu0 * sig2.0 + (n-1)*s.B + (k0*n*(ybar.B - mu0)^2)/(kn)) / nun

  # part B sampling
  s2.postsample.B = 1/rgamma(10000,nun/2, s2n.B*nun/2)
  theta.postsample.B = rnorm(10000, mun.B, sqrt(s2.postsample.B/kn))

  result = mean(theta.postsample.B > theta.postsample.A)

  probs = c(probs,result)
}

plot(c(1,2,4,8,16,32),probs,type='l')
```



6.1

a)

Since θ_B depends on θ_A, they are dependent. This might be warranted if you were trying to find a posterior for something like fish deaths in a river with priors on air pollution and water temperature. Temperature could be affected by air pollution, since temperature depends on pollution levels, so it makes sense that you would have dependent priors.

b)

Since θ and γ are independent, we can split the joint density below:

$$p(\theta, y_A, y_B, \gamma) \propto p(\theta)p(\gamma)p(y_A|\theta)p(y_B|\gamma \cdot \theta)$$

Now using the Poison sampling models and gamma priors, we end up with:

$$\begin{aligned} p(\theta|y_A, y_B, \gamma) &\propto c \cdot \theta^{a_\theta - 1 + \sum y_{i_A} + \sum y_{i_B}} \cdot e^{-\theta(b_\theta + n_A + \gamma n_B)} \\ &\propto gamma(a_\theta + \sum y_{i_A} + \sum y_{i_B}, b_\theta + n_A + \gamma n_B) \end{aligned}$$

c)

Proceeding the same as part b):

$$
\begin{aligned}
p(\gamma|y_A, y_B, \theta) &\propto c \cdot \gamma^{a_\gamma - 1 + \sum y_{i_B}} e^{-\gamma(b_\gamma + \theta n_B)} \\
&\propto gamma(a_\gamma + \sum y_{i_B}, b_\gamma + \theta \cdot n_B)
\end{aligned}
$$

d)

```
# loading data
# referenced this: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/scan

bach = scan(url('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/menchild30bach.dat'))
nobach =  scan(url('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/menchild30nobach.dat'))
```

Important: A represents with bachelors and B is without

We have:

$$
\begin{aligned}
p(\theta|everything) &= gamma(2 + 54 + 305, 1 + 58 + \gamma \cdot 218) = gamma(361, 59 + \gamma \cdot 218) \\
p(\gamma|everything) &= gamma(a_\gamma + 305, b_\gamma + \theta \cdot 218) \\
\theta_A &= \theta \\
\theta_B &= \theta \cdot \gamma
\end{aligned}
$$

Goal:

$$
E[\theta_B - \theta_A | y_A, y_B]
$$

Code below was adapted from lecture 10 p.28 and PH p. 94

```
# iterations for a_gamma and b_gamma
a.vals = c(8,16,32,64,128)
S=5000

# data
nA = length(bach)
nB = length(nobach)
yA.sum = sum(bach)
yB.sum = sum(nobach)
```

```r
theta = matrix(NA,nrow=S,ncol=2,dimnames=list(1:S,c("theta","gamma")))
theta.init = c(0,0)
theta.curr = theta.init

for (i in a.vals) {

  for (j in 1:S) {

    t = theta.curr[1]
    g = theta.curr[2]

    #update theta
    theta.curr[1] = rgamma(1,361,59 + g * 218)

    #update gamma
    theta.curr[2] = rgamma(1,i + 305, i + 218 * t)

    # saving current iteration
    theta[j,] = theta.curr
  }

  theta.A = theta[,1]
  theta.B = theta[,1] * theta[,2]

  E = mean(theta.B - theta.A)
  cat("E(thetaB - thetaA) for a_gamma = b_gamma = ", i, ":", E, "\n")
}
```

```
E(thetaB - thetaA) for a_gamma = b_gamma =  8 : 0.4695007
E(thetaB - thetaA) for a_gamma = b_gamma =  16 : 0.3527968
E(thetaB - thetaA) for a_gamma = b_gamma =  32 : 0.284862
E(thetaB - thetaA) for a_gamma = b_gamma =  64 : 0.2139508
E(thetaB - thetaA) for a_gamma = b_gamma =  128 : 0.1384996
```