Stat 572 Final Exam

Mark Roes

December 4, 2017

1.

Analytics

From section 5.4 in the text:

$p(\theta,\mu,\tau|y) \propto \prod_{j=1}^{n} N(\theta_j|\mu,\tau^2) \prod_{i=1}^{8} N(y_{ij}|\theta_j,\sigma^2 = .72)$

$\theta_j|\mu,\tau,y \sim N(\hat{\theta}_j, V_j)$

Where $\hat{\theta}_j = \frac{\frac{1}{.72}\bar{y}_{.j} + \frac{1}{\tau^2}\mu}{\frac{1}{.72}+\frac{1}{\tau^2}}$

$V_j = \frac{1}{\frac{1}{.72}+\frac{1}{\tau^2}}$

$\mu|\tau,y \sim N(\frac{\sum \frac{1}{.72+\tau^2}\bar{y}_j}{\sum \frac{1}{.72+\tau^2}}, \left(\sum \frac{1}{.72+\tau^2}\right)^{-1})$

$p(\tau|y) \propto \frac{p(\tau)\prod_{j=1}^{6} N(\bar{y}_j|\hat{\mu},.72+\tau^2)}{N(\hat{\mu}|\hat{\mu},V_u)}$

Where $\hat{\mu} = \frac{\sum \frac{1}{.72+\tau^2}\bar{y}_j}{\sum \frac{1}{.72+\tau^2}}$

Model

Found very little to no autocorrelation of parameters. No need to thin.

Found a burn-in period of about 150 for all parameters, will set to 200 to be safe.

A. Estimate for each population mean.

	node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
3	theta[1]	23.61	0.2951	23.02	23.41	23.61	23.81	24.20
4	theta[2]	10.72	0.3025	10.11	10.52	10.71	10.92	11.32
5	theta[3]	19.32	0.3013	18.75	19.11	19.33	19.53	19.90
6	theta[4]	17.49	0.2975	16.91	17.29	17.49	17.69	18.09
7	theta[5]	14.22	0.2971	13.62	14.03	14.23	14.43	14.79
8	theta[6]	20.40	0.3034	19.83	20.19	20.40	20.61	20.99

B. Estimate for the overall mean.

Using a pooled estimate for θ, I obtain an overall mean described in the table below:

node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
theta	17.65	0.1322	17.42	17.55	17.66	17.74	17.86

C. 95% credible interval for each population mean.

For all the 95% credible intervals, I used highest posterior density, as it gives a better estimate of the true density of our parameters.

Using HPD:

```
## [1] "HPD for theta"
## [1] 1
## lower upper
## 22.98 24.13
## [1] "HPD for theta"
## [1] 2
## lower upper
## 10.13 11.31
## [1] "HPD for theta"
## [1] 3
## lower upper
## 18.77 19.90
## [1] "HPD for theta"
## [1] 4
## lower upper
## 16.88 18.04
## [1] "HPD for theta"
## [1] 5
## lower upper
## 13.65 14.81
## [1] "HPD for theta"
## [1] 6
## lower upper
## 19.85 21.00
```

D. Ordering by sodium content.

Using the HPD calculations from part (c), I can order the beer brands by sodium content with minimal overlap of 95% intervals. In fact, the only overlap of these intervals is of $\theta_6 = [19.85, 21]$ and $\theta_3 = [18.77, 19.9]$. Considering there is only an overlap of 0.05 between the lower bound of θ_6 and the upper bound of θ_3, I still confidently state that $P(\theta_6 > 19.85) > P(\theta_3 > 19.85)$. I order from greatest to least sodium content as follows:

$\theta_1 > \theta_6 > \theta_3 > \theta_4 > \theta_5 > \theta_2$

CODE FOR Q1

```
#WinBUGS code
1. MODEL
model{
    for(j in 1:k){
        for(i in 1:n){
        y[i,j]~dnorm(theta[j],1.3889)

        }
    theta[j]~dnorm(mu,theta.tau)
    }
    mu~dnorm(0,.00001)
    recip.tau~dunif(0,1000)
    theta.tau<-pow(recip.tau,2)
```

```
    tau<-pow(recip.tau,-1)
}


MODEL FOR PART B
model{
    for(j in 1:k){
        for(i in 1:n){
        y[i,j]~dnorm(theta,1.3889)
        }
    }
    theta~dnorm(mu,theta.tau)
    mu~dnorm(0,.00001)
    recip.tau~dunif(0,1000)
    theta.tau<-pow(recip.tau,2)
    tau<-pow(recip.tau,-1)
}
DATA TRANSPOSE: COLS ARE BRANDS, ROWS ARE OBS
list(n=8,k=6, y = structure( .Data=c(24.4,10.2,19.2,17.4,13.4,
  21.3,22.6,12.1,19.4,18.1,15,20.2,23.8,10.3,19.8,16.7,14.1,20.7,
  22,10.2,19,18.3,13.1,20.8,24.5,9.9,19.6,17.6,14.9,20.1,22.3,11.2,
  18.3,17.5,15,18.8,25,12,20,18,13.4,21.1,24.5,9.5,19.4,16.4,14.8,20.3), .Dim=c(8,6)))

# R code
q1a<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq1stat.csv")
kable(q1a[-c(1,2),-c(4,10,11)])

q1b<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq1b.csv")
kable(q1b[,-c(4,10,11)])

q1cc<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq1coda.csv")
q1c<-cbind(q1cc[5603:8403,],q1cc[8404:11204,],q1cc[11205:14005,],
           q1cc[14006:16806,],q1cc[16807:19607,],q1cc[19608:22408,])

ic<-6
for(i in 1:ic){
  print("HPD for theta")
  print(i)
  print(hdi(q1c[,i])[1:2])
}
```

2.

Analytics

Going to call α, β as a,b to not confuse β, β_i $p(a,b,\sigma,\beta_i|y) \propto p(a,b)p(\beta_i|a,b)p(y|\beta_i,\sigma) = p(\beta_i|a,b)p(y|\beta_i,\sigma) = \prod_0^2 N(a,b) \prod_1^{40} N(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \sigma^2)$
$p(\beta_i|a,b,y) = \int_b \int_a \prod_0^2 N(a,b) \prod_1^{40} N(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \sigma^2) db\ da$
$p(\phi|y) = \prod^2 N(a,b) \int_{\beta_0} \int_{\beta_1} \int_{\beta_2} \int_\sigma \prod^{40} N(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \sigma^2) d\beta_0\ d\beta_1\ d\beta_2\ d\sigma$

Simulation

Very high autocorrelation out to about 15 lags. Going to thin by 15, as it appears there is no difference in autocorrelation if I thin by greater than this amount.

Given my initial values, not a long burn in period. Will burn in by 100 iterations.

A. Report 95% credible intervals for β_i

Using HPD:

```
## [1] "95% HPD for b0"

##  lower  upper
## 0.9422 4.9320

## [1] "95% HPD for b1"

##    lower    upper
## 0.009262 1.747000

## [1] "95% HPD for b2"

##    lower    upper
## 0.008623 0.020530
```

B. Find the best linear regression model fit to the data.

To find the the best model, I will assess the sum of squared error for the models:
$\beta_0 + \beta_1 x_1 + \beta_2 x_2$
$\beta_0 + \beta_1 x_1$
$\beta_0 + \beta_2 x_2$
β_0

Below are the distributions of sum of squared error for the 4 models.

Model	mean	sd	X2.5.	X25.0.	median	X75.0.	X97.5.
b0+b1x1+b2x2	570.0	43.34	523.10	538.7	558.5	590.7	682.6
b0+b1x1	913.6	56.52	842.60	867.7	906.5	944.8	1048.0
b0+b2x2	2333.0	191.20	15.51	2151.0	2191.0	2261.0	2417.0
b0	9610.0	366.70	9349.00	9378.0	9483.0	9689.0	10640.0

The mean of the distribution of SSE for the full model is much lower than the means of the other model SSEs. Lowering the number of parameters will increase our degrees of freedom, but will not make enough of a difference for any of the reduced models to surpass the full model.

Another argument is that the 95% HPDs for all the β_i are positive, so we ought to include them all.

I accept the full linear regression model: $\beta_0 + \beta_1 x_1 + \beta_2 x_2$ as the best model.

C. Using the model from (a), find a posterior predictive distribution of y for $x_1 = 11, x_2 = 200$

node	mean	sd	X2.5.	X25.0.	median	X75.0.	X97.5.
y.tilde	22.5	4.087	14.6	19.64	22.52	25.35	30.51

D. Find the posterior distribution of the sum of squared error.

node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
sumsqerr	570	43.34	523.1	538.7	558.5	590.7	682.6

Code Used:

```
#WinBUGS code
2. MODEL
model{
    for(i in 1:n){
    y[i]~dnorm(theta[i],pre.sig)
    theta[i]<-b0+b1*x1[i]+b2*x2[i]

    err[i]<-theta[i]-y[i]
    sqerr[i]<-pow(err[i],2)

    }
alpha~dnorm(0,pre.alpha)
pre.alpha<-pow(1000,-1)

pre.sig~dgamma(.001,.001)
real.v<-sqrt(pow(pre.sig,-1))

b0~dnorm(alpha,beta)
b1~dnorm(alpha,beta)
b2~dnorm(alpha,beta)

theta.tilde<-b0+b1*11+b2*200

y.tilde~dnorm(theta.tilde,pre.sig)
beta~dgamma(.001,.001)
sumsqerr<-sum(sqerr[])
}

DATA
list(n=40,y=c(16.68,11.5,12.03,14.88,13.75,18.11,8,
              17.83,79.24,21.5,40.33,21,13.5,19.75, 24,29,15.35,
              19,9.5,35.1,17.9,52.32,18.75,19.83,10.75,51,
              16.8,26.16,19.9,24,18.55,31.93,6.95,7,14,37.03,18.62,
              15.10,24.38,64.75),

x1=c(7,3,3,4,6,7,2,7,30,5,
16,10,4,6,9,10,6,7,3,17,
10,26,9,8,4,22,7,15,5,6,6,
10,4,1,3,12,10,7,8,32),

x2=c(560,220,340,80,150,330,
110,210,1460,605,688,215,255,
462,448,776,200,132,36,770,140,
810,450,635,150,905,520,290,500,
1000,225,775,212,144,126,655,420,
150,360,1530))

INITS
```

```
list(pre.sig=.01, alpha=5, beta=2.4,b0=1,b1=1,b2=1,y.tilde=1)


MODEL part b-1 (b0+b1x1)
model{
    for(i in 1:n){
    y[i]~dnorm(theta[i],pre.sig)
    theta[i]<-b0+b1*x1[i]

    err[i]<-theta[i]-y[i]
    sqerr[i]<-pow(err[i],2)

    }
alpha~dnorm(0,pre.alpha)
pre.alpha<-pow(1000,-1)

pre.sig~dgamma(.001,.001)
beta~dgamma(.001,.001)
b0~dnorm(alpha,beta)
b1~dnorm(alpha,beta)
sumsqerr<-sum(sqerr[])
}
DATA
list(n=40,y=c(16.68,11.5,12.03,14.88,13.75,18.11,8,17.83,
              79.24,21.5,40.33,21,13.5,19.75,24,29,15.35,19,
              9.5,35.1,17.9,52.32,18.75,19.83,10.75,51,16.8,
              26.16,19.9,24,18.55,31.93,6.95,7,14,37.03,18.62,
              15.10,24.38,64.75),

x1=c(7,3,3,4,6,7,2,7,30,5,
16,10,4,6,9,10,6,7,3,17,
10,26,9,8,4,22,7,15,5,6,6,
10,4,1,3,12,10,7,8,32))

INITS
list(pre.sig=.01, alpha=5, beta=2.4,b0=1,b1=1)

MODEL part b-2 (b0+b2x2)
model{
    for(i in 1:n){
    y[i]~dnorm(theta[i],pre.sig)
    theta[i]<-b0+b2*x2[i]

    err[i]<-theta[i]-y[i]
    sqerr[i]<-pow(err[i],2)

    }
alpha~dnorm(0,pre.alpha)
pre.alpha<-pow(1000,-1)

pre.sig~dgamma(.001,.001)
beta~dgamma(.001,.001)
b0~dnorm(alpha,beta)
```

```
b2~dnorm(alpha,beta)
sumsqerr<-sum(sqerr[])
}

DATA
list(n=40,y=c(16.68,11.5,12.03,14.88,13.75,18.11,
              8,17.83,79.24,21.5,40.33,21,13.5,19.75, 24,
              29,15.35,19,9.5,35.1,17.9,52.32,18.75,19.83,10.75,
              51,16.8,26.16,19.9,24,18.55,31.93,6.95,7,14,37.03,
              18.62,15.10,24.38,64.75),

x2=c(560,220,340,80,150,330,
110,210,1460,605,688,215,255,
462,448,776,200,132,36,770,140,
810,450,635,150,905,520,290,500,
1000,225,775,212,144,126,655,420,
150,360,1530))

INITS
list(pre.sig=.01, alpha=5, beta=2.4,b0=1,b2=1)

MODEL part b-3 (b0)
model{
    for(i in 1:n){
    y[i]~dnorm(theta[i],pre.sig)
    theta[i]<-b0

    err[i]<-theta[i]-y[i]
    sqerr[i]<-pow(err[i],2)

    }
alpha~dnorm(0,pre.alpha)
pre.alpha<-pow(1000,-1)

pre.sig~dgamma(.001,.001)
beta~dgamma(.001,.001)
b0~dnorm(alpha,beta)
sumsqerr<-sum(sqerr[])
}
DATA
list(n=40,y=c(16.68,11.5,12.03,14.88,13.75,18.11,8,
              17.83,79.24,21.5,40.33,21,13.5,19.75,
              24,29,15.35,19,9.5,35.1,17.9,52.32,18.75,19.83,10.75,
              51,16.8,26.16,19.9,24,18.55,31.93,6.95,7,14,37.03,18.
              62,15.10,24.38,64.75))

INITS
list(pre.sig=.01, alpha=5, beta=2.4,b0=1)



# R Code
q2stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq2stat.csv")
```

```
q2d<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq2coda.csv")
q2data<-cbind(q2d[1:1901,],q2d[1902:3902,],q2d[3803:5703,])
print("95% HPD for b0")
hdi(q2data[,1])[1:2]
print("95% HPD for b1")
hdi(q2data[,2])[1:2]
print("95% HPD for b2")
hdi(q2data[,3])[1:2]

q2stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq2b.csv")
kable(q2stat)

q2cstat<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq2cstat.csv")
kable(q2cstat)

q2ds<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq2dstat.csv")
kable(q2ds)
```

3.

Since μ_A, μ_B are from the same distribution, we can cover both of them without a loss of generality.

$p(\mu|y) = \frac{p(y|\mu)p(\mu)}{p(y)} \propto \prod N(\mu, 6^2)N(100, 20^2)$

$= \exp\left[\sum_{y_j} -\frac{1}{2(6^2)}(y_j - \mu)^2\right] \exp\left[\frac{1}{2(20^2)}(\mu - 100)^2\right]$

$= \exp\left[\sum_{y_j} -\frac{1}{2(6^2)}(y_j - \mu)^2 - \frac{1}{2(20^2)}(\mu - 100)^2\right]$

$= \exp\left[-\frac{1}{2(6^2)}(\sum y_j + \sum \mu^2 - 2\mu \sum y_j) - \frac{1}{2(20^2)}(\mu^2 + 100^2 - 200\mu)\right]$

$\propto \exp\left[-\frac{1}{2(6^2)}(10\mu^2 + 2\mu \sum y_j) - \frac{1}{2(20^2)}(\mu^2 + 200\mu)\right]$

There may be a closed form if I did a clever transformation on these variables, but I will leave this as is.

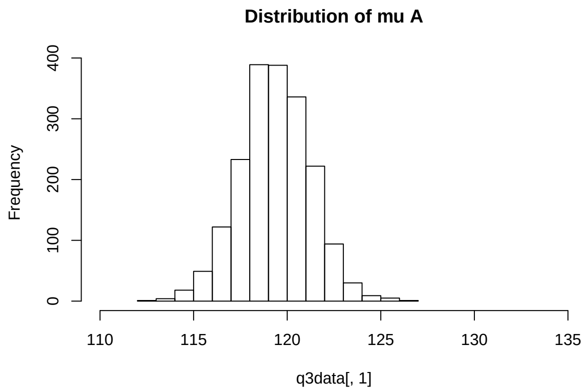
Simulations

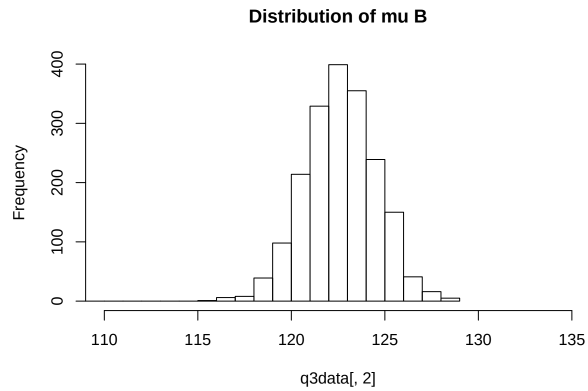
Found very low autocorrelation, no necessity to thin. Also a very short burn-in period, setting burn in time to 100.

A. Find the posterior distributions of μ_a, μ_b

node	mean	sd	X2.5.	X25.0.	median	X75.0.	X97.5.
mu_a	119.4	1.868	115.6	118.2	119.5	120.7	123.0
mu_b	122.7	1.902	118.9	121.4	122.7	124.0	126.4

And visually:



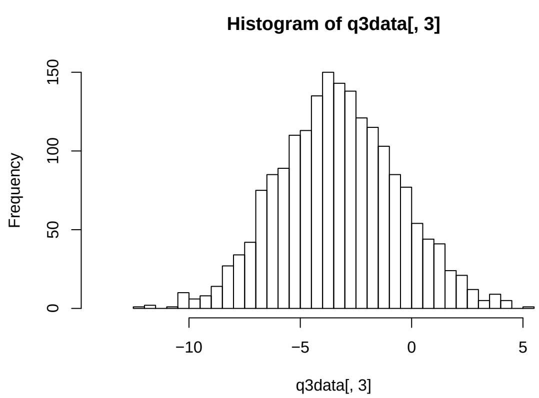


B. Find the posterior Distribution of $\mu_A - \mu_B$

Here are the basic stats for $\mu_A - \mu_B$

	node	mean	sd	X2.5.	X25.0.	median	X75.0.	X97.5.
3	partb	-3.278	2.689	-8.289	-5.127	-3.323	-1.477	2.179

And visually:



C. Find a 95% Bayesian credible interval for $\mu_A - \mu_B$

Using HPD:

```
##  lower  upper
## -8.222  2.203
```

D. Which method would you prefer to use?

Given the median and histogram from part (b) and the credible interval from (c), it is clear that method A is estimated to be quicker than method B. $\mu_A - \mu_B$ has most of its distribution below zero, therefore $P(\mu_B > \mu_A) > 1/2$. This means that applying method B generally takes longer to complete the task than method A. I recommend method A.

Code Used:

```
#WinBUGS code
3. MODEL
model{
    for(i in 1:n){
    a[i]~dnorm(mu_a,pre)
    b[i]~dnorm(mu_b,pre)

    }
mu_a~dnorm(100,pres)
mu_b~dnorm(100,pres)
pres<-pow(20,-2)
pre<-pow(6,-2)
partb<-mu_a-mu_b
}
DATA
list(n=10,a=c(115,120,111,123,116,121,118,116,127,129),
     b=c(123,131,113,119,123,113,128,126,125,128))

#R code
q3stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq3stat.csv")
q3d<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq3coda.csv")[,2]
q3data<-cbind(q3d[1:1901],q3d[1902:3802],q3d[3803:5703])
kable(q3stat[-3,-4])

hist(q3data[,1], main = "Distribution of mu A", xlim=c(110,135), breaks =15)
hist(q3data[,2], main = "Distribution of mu B", xlim= c(110,135), breaks = 100)

kable(q3stat[3,-4])

hist(q3data[,3], xlim=c(-14,7), breaks=50)

hdi(q3data[,3])[1:2]
```

4.

Analytics.

Model 1.
$p(\alpha|y) \propto p(\alpha)p(y|\alpha) = exponential(1)\prod_{i=1}^{10} Poisson(\alpha + \log(t_i))$

Model 2.
$p(\alpha, \beta|y) \propto p(\alpha)p(\beta)p(y|\alpha, \beta) = exponential(1)Gamma(.1, 1)\prod^{10} poisson(\alpha + \beta(\log(t_i)))$

Model 3.
$p(\theta_i|y) = p(\theta_i)p(y|\theta_i) = Gamma(.1, 1)\prod^{10} Poisson(\theta_i t_i)$

Model 1.

Found that autocorrelation for λ_i and α was almost zero, so no need to thin.

Burn in period found to be quite short, via checking the trace. Setting burn in time to 100.

Posterior distribution of the parameters:

node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
a	3.605e-03	0.00360	7.290e-05	1.047e-03	0.00251	5.056e-03	0.0132
lam[1]	9.484e+01	0.34300	9.451e+01	9.460e+01	94.74000	9.498e+01	95.7600
lam[2]	1.576e+01	0.05699	1.570e+01	1.572e+01	15.74000	1.578e+01	15.9100
lam[3]	6.313e+01	0.22830	6.290e+01	6.297e+01	63.06000	6.322e+01	63.7400
lam[4]	1.265e+02	0.45740	1.260e+02	1.261e+02	126.30000	1.266e+02	127.7000
lam[5]	5.259e+00	0.01900	5.240e+00	5.245e+00	5.25300	5.267e+00	5.3100
lam[6]	3.151e+01	0.11400	3.140e+01	3.143e+01	31.48000	3.156e+01	31.8200
lam[7]	1.054e+00	0.00381	1.050e+00	1.051e+00	1.05300	1.055e+00	1.0640
lam[8]	1.054e+00	0.00381	1.050e+00	1.051e+00	1.05300	1.055e+00	1.0640
lam[9]	2.108e+00	0.00762	2.100e+00	2.102e+00	2.10500	2.111e+00	2.1280
lam[10]	1.054e+01	0.03810	1.050e+01	1.051e+01	10.53000	1.055e+01	10.6400

```
## [1] "HPD for a"
```

```
##     lower     upper
## 2.720e-06 1.061e-02
```

```
## [1] "HPD for lambda"
## [1] 1
## lower upper
## 94.50 95.51
## [1] "HPD for lambda"
## [1] 2
## lower upper
## 15.70 15.87
## [1] "HPD for lambda"
## [1] 3
## lower upper
## 62.90 63.57
## [1] "HPD for lambda"
## [1] 4
## lower upper
## 126.0 127.3
## [1] "HPD for lambda"
## [1] 5
## lower upper
## 5.240 5.296
## [1] "HPD for lambda"
## [1] 6
## lower upper
## 31.40 31.73
## [1] "HPD for lambda"
## [1] 7
## lower upper
## 1.050 1.061
## [1] "HPD for lambda"
## [1] 8
## lower upper
## 1.050 1.061
## [1] "HPD for lambda"
## [1] 9
## lower upper
## 2.100 2.122
```

```
## [1] "HPD for lambda"
## [1] 10
## lower upper
## 10.50 10.61
```

Above: upper and lower bounds for 95% credible intervals using the hpd method for a, λ_i, listed sequentially with a first, and λ from 1 to 10.

Model 2.

Found that there is higher autocorrelation in this model. Therefore I will thin the data by 10 and re-test. As you can see below, lagging the model summarily dealt with the autocorrelation.

The trace also indicates there is a very short (if any) burn in period. So I will choose 100 again.

Posterior distribution of the parameters:

node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
a	1.2510	0.2737	0.7111	1.0680	1.2630	1.4460	1.7540
b	0.2597	0.0787	0.1078	0.2064	0.2607	0.3132	0.4157
lam[1]	11.5500	1.9050	8.0310	10.2400	11.4700	12.7800	15.5900
lam[2]	7.2030	0.9056	5.5950	6.5520	7.1490	7.7960	9.1250
lam[3]	10.3600	1.5090	7.5680	9.3170	10.3000	11.3400	13.4400
lam[4]	12.4800	2.2570	8.3430	10.9500	12.3800	13.9400	17.3400
lam[5]	5.4500	0.9089	3.8700	4.8030	5.3950	6.0200	7.3930
lam[6]	8.6240	1.0780	6.6660	7.9010	8.5810	9.2920	10.9000
lam[7]	3.6700	0.9886	2.0760	2.9530	3.5790	4.2890	5.8140
lam[8]	3.6700	0.9886	2.0760	2.9530	3.5790	4.2890	5.8140
lam[9]	4.3440	0.9628	2.7260	3.6380	4.2700	4.9540	6.4480
lam[10]	6.4980	0.8844	4.9000	5.8720	6.4730	7.0410	8.3820

```
## [1] "HPD for a"

##  lower  upper
## 0.7338 1.7670

## [1] "HPD for b"

##  lower  upper
## 0.1121 0.4187

## [1] "HPD for lambda"
## [1] 1
## lower upper
## 15.70 15.87
## [1] "HPD for lambda"
## [1] 2
## lower upper
## 5.628 9.128
## [1] "HPD for lambda"
## [1] 3
##  lower  upper
##  7.326 13.140
## [1] "HPD for lambda"
## [1] 4
##  lower  upper
```

```
##  8.205 17.070
## [1] "HPD for lambda"
## [1] 5
## lower upper
## 3.673 7.126
## [1] "HPD for lambda"
## [1] 6
##  lower  upper
##  6.666 10.890
## [1] "HPD for lambda"
## [1] 7
## lower upper
## 1.926 5.616
## [1] "HPD for lambda"
## [1] 8
## lower upper
## 1.926 5.616
## [1] "HPD for lambda"
## [1] 9
## lower upper
## 2.505 6.121
## [1] "HPD for lambda"
## [1] 10
## lower upper
## 4.854 8.252
```

Above: upper and lower bounds for 95% credible intervals using the hpd method for a, b, λ_i. They are listed in this order.

Model 3.

Found very little autocorrelation in the λ and θ values of the model. So no thinning necessary.

As with the other two models there is very little, if any burn in period. So to keep consistent, will stick with 100.

For the posterior distribution of the parameters, I am only concerned with λ_i, as θ_i is λ_i scaled by the value at t_i.

node	mean	sd	X2.50.	X25.00.	median	X75.00.	X97.50.
lam[1]	5.0230	2.2840	1.65200	3.3850	4.6940	6.2670	10.520
lam[2]	0.9924	0.9536	0.04227	0.3112	0.7039	1.4010	3.498
lam[3]	5.0200	2.1710	1.76600	3.3610	4.7080	6.3080	10.020
lam[4]	14.0500	3.7640	7.71400	11.3800	13.7200	16.2400	22.560
lam[5]	2.6040	1.4620	0.55930	1.5100	2.3510	3.4260	6.098
lam[6]	18.3700	4.2550	11.12000	15.3500	17.9800	21.0200	27.910
lam[7]	0.5689	0.5308	0.01736	0.1866	0.4177	0.7963	1.910
lam[8]	0.5709	0.5467	0.02121	0.1765	0.4063	0.7993	1.983
lam[9]	2.7990	1.3630	0.80100	1.7970	2.5650	3.5900	5.849
lam[10]	20.1900	4.3870	12.65000	17.0100	19.7900	22.9100	29.590

```
## [1] "HDI for lambda"
## [1] 1
## lower upper
```

```
## 1.329 9.692
## [1] "HDI for lambda"
## [1] 2
##    lower    upper
## 0.000833 2.840000
## [1] "HDI for lambda"
## [1] 3
## lower upper
## 1.469 9.337
## [1] "HDI for lambda"
## [1] 4
##  lower  upper
##  7.167 21.720
## [1] "HDI for lambda"
## [1] 5
##  lower  upper
## 0.2002 5.4130
## [1] "HDI for lambda"
## [1] 6
## lower upper
## 10.70 26.87
## [1] "HDI for lambda"
## [1] 7
##    lower    upper
## 0.000673 1.599000
## [1] "HDI for lambda"
## [1] 8
##    lower    upper
## 0.000642 1.653000
## [1] "HDI for lambda"
## [1] 9
##  lower  upper
## 0.5699 5.3690
## [1] "HDI for lambda"
## [1] 10
## lower upper
## 11.96 28.59
```

Above: upper and lower bounds for 95% credible intervals using the hpd method for λ_i, listed sequentially from 1 to 10.

B.

Using WinBUGS, I conducted posterior predicitions for all three models with the points $t_{11} = 15.7, t_{12} = 10$ added to the model. I will choose the model for which the λ_i has the narrowest 95% credible interval.

```
## [1] "95% HPDs for Model 1"
```

```
## [1] "lambda 11"
```

```
## lower upper
## 15.70 15.87
```

```
## [1] "lambda 12"
```

```
## lower upper
## 10.00 10.11
```

```
## [1] "95% HPDs for Model 2"

## [1] "lambda 11"

## lower upper
## 5.555 9.023

## [1] "lambda 12"

## lower upper
## 4.814 8.410

## [1] "95% HPDs for Model 3"

## [1] "lambda 11"

##     lower     upper
## 3.058e-36 9.166e+00

## [1] "lambda 12"

##     lower     upper
## 5.477e-40 5.831e+00
```

Between these models, model 1 has by far the narrowest 95% credible interval for $\lambda_{11}, \lambda_{12}$ So I will choose model 1 for both predictions.

Code Used:

```
#WinBUGS
4.
 MODEL A
model{
    for(i in 1:n){
    y[i]~dpois(lam[i])
    lam[i]<-exp(ln_lam[i])
    ln_lam[i]<-a+ln_t[i]
    ln_t[i]<-log(t[i])
    }
a~dexp(1)
}

MODEL B
model{
    for(i in 1:n){
    y[i]~dpois(lam[i])
    ln_t[i]<-log(t[i])
    ln_lam[i]<-a+b*ln_t[i]
    lam[i]<-exp(ln_lam[i])
    }
a~dexp(1)
b~dgamma(.1,1)
}


MODEL C
model{
    for(i in 1:n){
    y[i]~dpois(lam[i])
```

```
    lam[i]<-theta[i]*t[i]
    theta[i]~dgamma(.1,1)
    }
}

DATA
list(n=10,y=c(5,1,5,14,3,19,1,1,4,22),
     t=c(94.5,15.7,62.9,126,5.24,31.4,1.05,1.05,2.1,10.5))

DATA FOR PART B: PREDICTIVE
list(n=12,y=c(5,1,5,14,3,19,1,1,4,22,NA,NA),
     t=c(94.5,15.7,62.9,126,5.24,31.4,1.05,1.05,2.1,10.5,15.7,10))

#R Code
q4m1stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bayesfinalq4m1stat.csv")
kable(q4m1stat[,-c(4,10,11)])
q4m1<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4m1coda.csv")
m1df<-cbind(q4m1[1:1901,2],q4m1[1902:3802,2],
            q4m1[3803:5703,2],q4m1[5704:7604,2],
            q4m1[7605:9505,2],q4m1[9506:11406,2],
            q4m1[11407:13307,2],q4m1[13308:15208,2],
            q4m1[15209:17109,2],q4m1[17110:19010,2],
            q4m1[19011:20911,2])

u<-11
print("HPD for a")
hdi(m1df[,1])[1:2]
for(i in 2:u){
  print("HPD for lambda")
  print(i-1)
  print(hdi(m1df[,(i)])[1:2])
}


q4m2stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bayesfinalq4m2stat.csv")
kable(q4m2stat[,-c(4,10,11)])
q4m2<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4m2coda.csv")
m2df<-cbind(q4m2[1:1901,2],q4m2[1902:3802,2],q4m1[3803:5703,2],
            q4m2[5704:7604,2],q4m2[7605:9505,2],q4m2[9506:11406,2],
            q4m2[11407:13307,2],q4m2[13308:15208,2],q4m2[15209:17109,2],
            q4m2[17110:19010,2],q4m2[19011:20911,2],q4m2[20912:22812,2])

print("HPD for a")
hdi(m2df[,1])[1:2]
print("HPD for b")
hdi(m2df[,2])[1:2]
u2<-12
for(i in 3:u2){
  print("HPD for lambda")
  print(i-2)
  print(hdi(m2df[,(i)])[1:2])
}
```

```
q4m3stat<-read.csv("C:/Users/Mark/Documents/Stat 572/bayesfinalq4m3stat.csv")
kable(q4m3stat[-c(11:20),-c(4,10,11)])
q4m3<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4m3coda.csv")
m3df<-cbind(q4m3[1:1901,2],q4m3[1902:3802,2],q4m3[3803:5703,2],
            q4m3[5704:7604,2],q4m3[7605:9505,2],q4m3[9506:11406,2],
            q4m3[11407:13307,2],q4m3[13308:15208,2],q4m3[15209:17109,2],q4m3[17110:19010,2])

u3<-10
for(i in 1:u3){
  print("HDI for lambda")
  print(i)
  print(hdi(m3df[,(i)])[1:2])
}

q4b11<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4b1.csv")
q4b1<-cbind(q4b11[1:1901,],q4b11[1902:3802,])
print("95% HPDs for Model 1")
print("lambda 11")
hdi(q4b1[,1])[1:2]
print("lambda 12")
hdi(q4b1[,2])[1:2]

q4b22<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4b2.csv")
q4b2<-cbind(q4b22[1:1901,],q4b22[1902:3802,])
print("95% HPDs for Model 2")
print("lambda 11")
hdi(q4b2[,1])[1:2]
print("lambda 12")
hdi(q4b2[,2])[1:2]

q4b33<-read.csv("C:/Users/Mark/Documents/Stat 572/bfinalq4b3.csv")
q4b3<-cbind(q4b33[1:1901,],q4b33[1902:3802,])
print("95% HPDs for Model 3")
print("lambda 11")
hdi(q4b3[,1])[1:2]
print("lambda 12")
hdi(q4b3[,2])[1:2]
```