Question 1:

Coded the below OpenBUGS code:

```
secondbirth0-1

model{
for(i in 1:N){
time[i] ~ dnorm(mu[i], tau)
mu[i] <- b0 + b1 * mage[i] + b2*death[i]
}

b0 ~ dnorm(0, 0.001)
b1 ~ dnorm(0, 0.001)
b2 ~ dnorm(0, 0.001)
tau ~ dgamma(0.001, 0.001)

helgaMage <- 24
helgaDeath <- 0
helgaTime <- b0 + b1 * helgaMage + b2 * helgaDeath
helgaTimePredicted ~ dnorm (helgaTime, tau)

emmaMage <- 28
emmaDeath <- 1
emmaTime <- b0 + b1* emmaMage + b2*emmaDeath
emmaTimePredicted ~ dnorm (emmaTime, tau)

}

list(N=16341)

→| DATA(mage, death, and time) ←

INITS
list(b0=1, b1=0, b2=0, tau=1)
```

Got the below node statistics after burning 1000 samples:

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b0	1009.0	25.27	2.008	958.0	1011.0	1055.0	1001	10000
b1	-1.192	1.086	0.08641	-3.161	-1.276	1.006	1001	10000
b2	-171.2	23.44	0.2182	-217.0	-171.3	-125.1	1001	10000
deviance	244300.0	51.75	4.059	244200.0	244300.0	244400.0	1001	10000
emmaTimePredicted	799.6	430.5	4.604	-34.61	798.3	1659.0	1001	10000
helgaTimePredicted	978.6	424.8	4.075	139.7	976.6	1799.0	1001	10000
tau	5.498E-6	6.325E-8	1.535E-9	5.376E-6	5.498E-6	5.622E-6	1001	10000

Therefore, the Bayesian linear regression equation is:

time = 1009.0 -1.192 * mage -171.2 * death

(a) From the above node statistics, the 95% credible set for the slope b2 is **[-217.0, -125.1]**. Yes, b2 **is significant** because it has high negative value and it's 95% CS doesn't contain 0.

(b) The Bayesian estimate of b1 from node statistics is -1.192 which is not a very significant value as compared to b2. Also, the 95% credible set of b1 is [-3.161, 1.006] **which contains 0**. Therefore, we can say that b1 **is not a significant factor** in influencing the response time.

I also created a new model, to check how the regression equation changes if we explicitly put b1 = 0.

Found the below node statistics, after explicitly putting b1 = 0.

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b0	981.3	3.315	0.04128	974.7	981.2	987.8	1001	10000
b2	-170.7	23.47	0.2217	-216.8	-170.8	-124.6	1001	10000
deviance	244300.0	10.82	0.1323	244300.0	244300.0	244400.0	1001	10000
tau	5.479E-6	6.081E-8	6.999E-10	5.363E-6	5.479E-6	5.597E-6	1001	10000

Deviance measures in-sample accuracy of the model. From above, we can see that deviance is almost the same and there isn't much difference in the b2 and it's 95% CS after putting b1 = 0. Therefore, we can conclude that b1 is insignificant.

The Bayesian linear regression after ignoring b1 is:

time = 981.3 -170.7 * death

(c) The predicted time between the births for Helga is given by helgaTimePredicted in the node statistics. The Bayesian estimate for helgaTimePredicted is given by 978.6. Therefore, the predicted time between the births according to my model for Helga is **978.6**.

(d) The predicted time between the births for Emma is given by emmaTimePredicted in the node statistics. Therefore, the 95% CS for the predicted time between the births for Emma according to my model is given by: **[-34.61, 1659.0]**.

However, please not here that lower bound of the CS is -34.61 for time which is not possible in the practical world. The time should always be greater than or equal to 0. 0 would be the case for twins.

Question 2:

(a) Without interaction term. Used the below OpenBUGS code:

cloudsNoIntersectionTerm

```
model{
  for(i in 1:n){
  diff[i]  ~ dnorm( mu[i], tau )
   mu[i] <-  mu0 + alpha[ seeded[i] ]  + beta[ season[i]  ]
   }

##STZ (sum-to-zero) constraints
  alpha[1] <- - sum(alpha[2:leva])
  beta[1] <- - sum(beta[2:levb])
  for(a in 1:leva) {alpha.beta[a,1] <- - sum(alpha.beta[a, 2:levb])}
  for(b in  2:levb) {alpha.beta[1,b] <-  - sum(alpha.beta[2:leva, b])}

#PRIORS
mu0 ~ dnorm(0, 0.0001)
for(a in 2:leva) {alpha[a] ~ dnorm(0, 0.0001)}
for(b in 2:levb) {beta[b] ~ dnorm(0, 0.0001)}
for(a in 2:leva) {for(b in 2:levb){
    alpha.beta[a,b] ~ dnorm(0, 0.0001) }}
tau ~ dgamma(0.001, 0.001)
s <- 1/sqrt(tau)

#PAIRWISE COMPARISONS

for(i in 1:1)  {for(j in i+1:2) {ca[i,j] <- alpha[i]-alpha[j]}}
for(i in 1:3)  {for(j in i+1:4) {cb[i,j] <- beta[i]-beta[j]}}

}


DATA 1
list(n =108, leva= 2, levb= 4)

→ DATA2 ←


INITS
list(mu0=0, alpha=c(NA,  0), beta=c(NA,0,0,0), tau = 1)
```

There slope for seeded is alpha and the slope for season is beta.

Note that I used, seeded to be 1 and unseeded to be 2.

Note that I used Spring = 1, Summer = 2, Autumn = 3 and Winter = 4.

Node statistics obtained after burning 1000 samples:

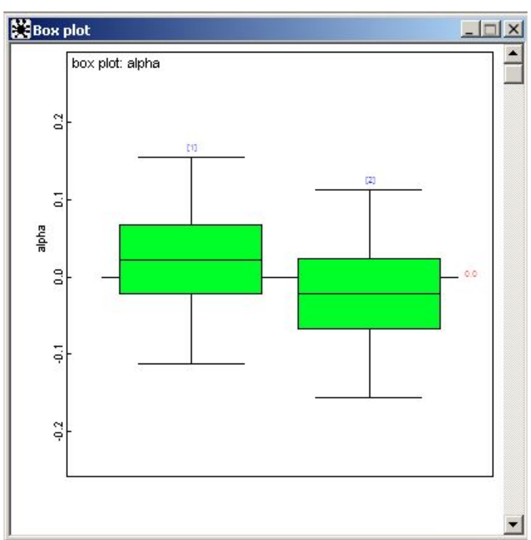
Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha[1]	0.02172	0.06837	2.135E-4	-0.113	0.02157	0.1554	1001	100000
alpha[2]	-0.02172	0.06837	2.135E-4	-0.1554	-0.02157	0.113	1001	100000
beta[1]	0.2033	0.1125	2.431E-4	-0.01605	0.2029	0.4253	1001	100000
beta[2]	0.2228	0.1242	5.517E-4	-0.02006	0.2231	0.4672	1001	100000
beta[3]	0.05813	0.1235	5.536E-4	-0.1834	0.05785	0.3029	1001	100000
beta[4]	-0.4842	0.1175	4.898E-4	-0.7144	-0.4841	-0.2522	1001	100000
ca[1,2]	0.04344	0.1367	4.27E-4	-0.2259	0.04314	0.3107	1001	100000
cb[1,2]	-0.01953	0.1925	6.473E-4	-0.3962	-0.01982	0.3566	1001	100000
cb[1,3]	0.1451	0.1919	6.441E-4	-0.2334	0.145	0.5204	1001	100000
cb[1,4]	0.6874	0.1841	5.589E-4	0.3269	0.6863	1.051	1001	100000
cb[2,3]	0.1647	0.2056	9.674E-4	-0.2401	0.1653	0.5675	1001	100000
cb[2,4]	0.707	0.1987	8.788E-4	0.3162	0.7073	1.097	1001	100000
cb[3,4]	0.5423	0.1976	8.859E-4	0.1541	0.542	0.9325	1001	100000
deviance	231.7	3.592	0.01343	226.8	231.0	240.5	1001	100000
s	0.7089	0.05024	1.597E-4	0.6187	0.706	0.816	1001	100000
tau	2.02	0.2829	8.793E-4	1.502	2.006	2.612	1001	100000

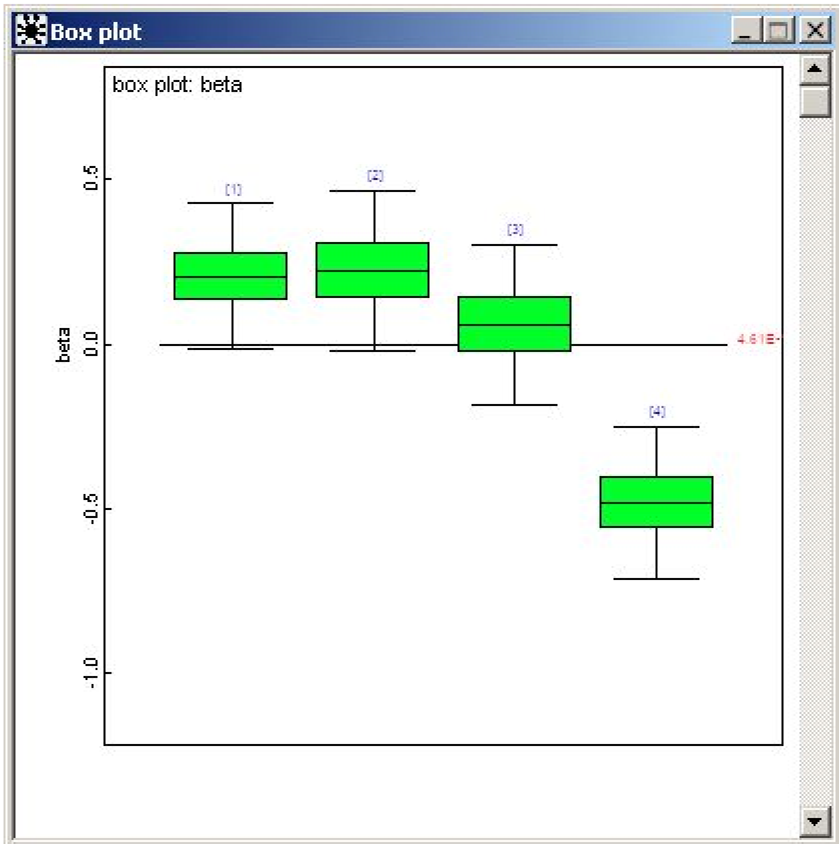
Since 0 is contained in the 95% CS: ca[1,2]; cb[1,2]; cb[1,3]; cb[2,3]; they are not different.

Similarly, we can look above in all the 95% CS for all the comparisons and can infer. If 0 is contained in the credible set, they are not different; Otherwise, if 0 is not contained in 95% CS, they are different.

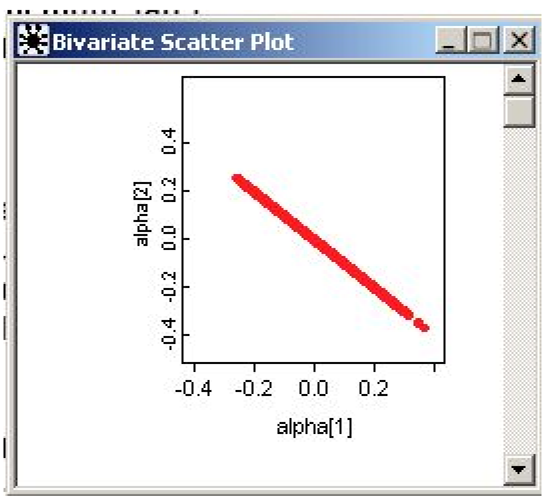
Boxplot for alpha:



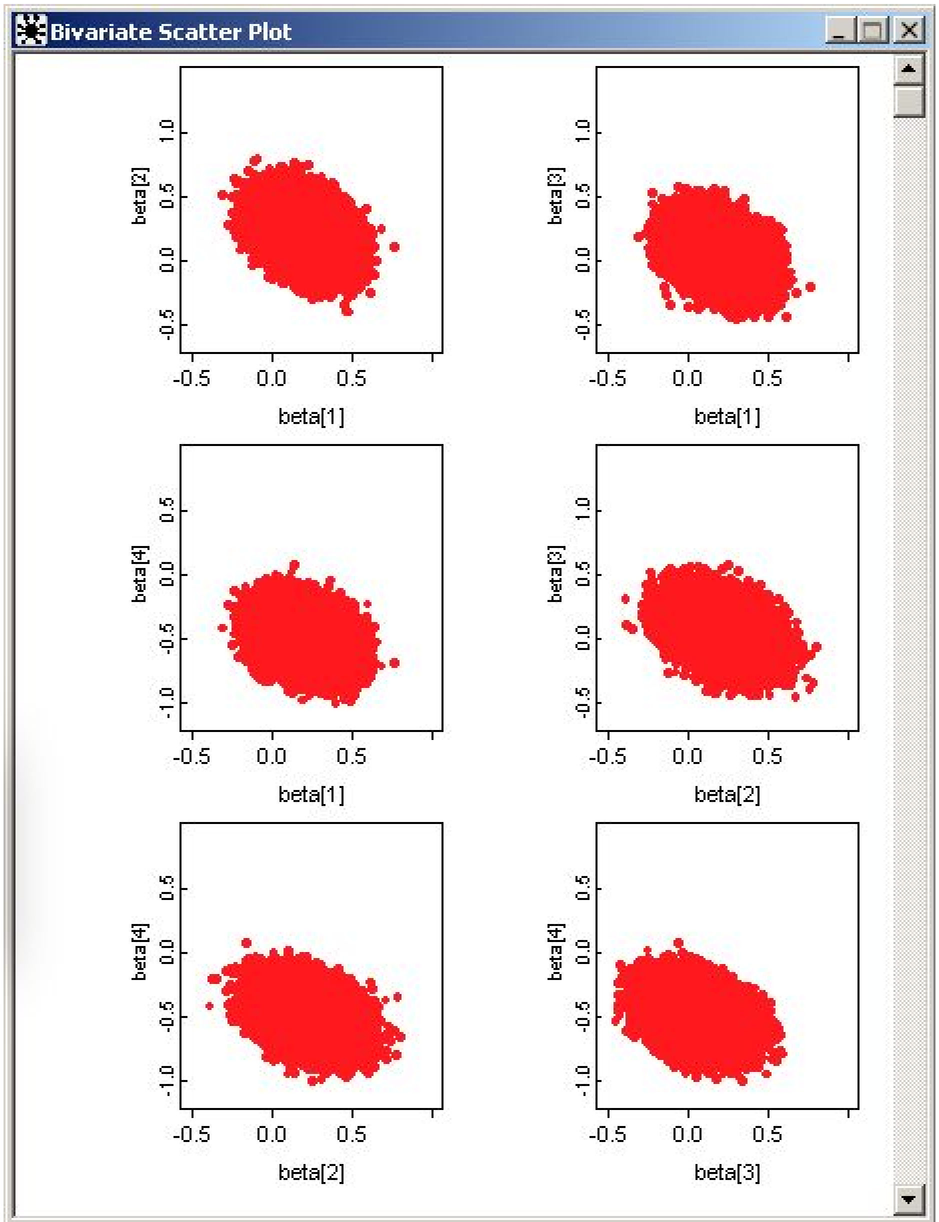
Boxplot for Beta:



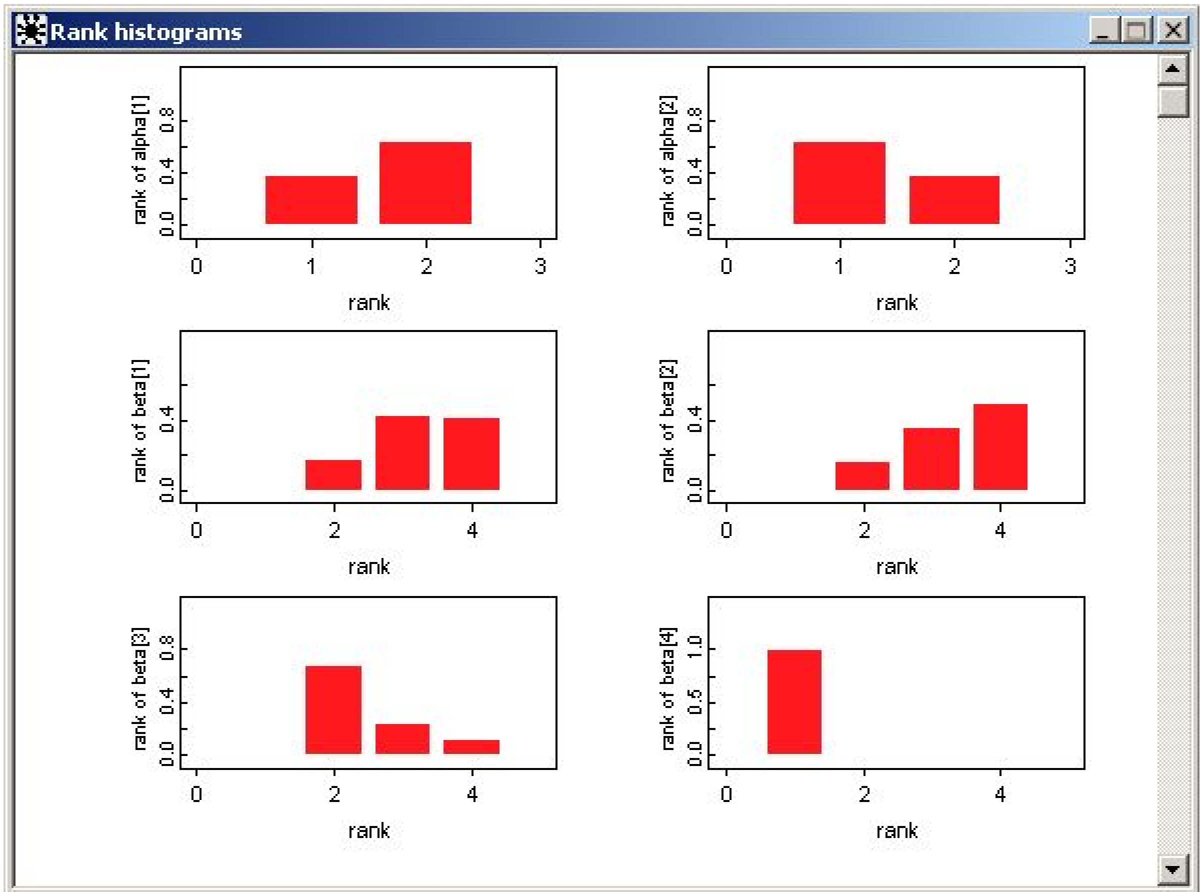
Correlation alpha:



Correlation beta:



Rank histograms:



Rank probability: alpha[1] > alpha[1]

Rank probability: beta[2] > beta[1] > beta[3] > beta[4]

(b) With interaction term. Used the below OpenBUGS code:

There slope for seeded is alpha and the slope for season is beta.

Note that I used, seeded to be 1 and unseeded to be 2.

Note that I used Spring = 1, Summer = 2, Autumn = 3 and Winter = 4.

clouds

```
model{
  for(i in 1:n){
  diff[i]  ~ dnorm( mu[i], tau )
   mu[i] <-  mu0 + alpha[ seeded[i] ]  + beta[ season[i]  ] + alpha.beta[ seeded[i], season[i]
]
   }

##STZ (sum-to-zero) constraints
  alpha[1] <- - sum(alpha[2:leva])
  beta[1] <- - sum(beta[2:levb])
  for(a in 1:leva) {alpha.beta[a,1] <- - sum(alpha.beta[a, 2:levb])}
  for(b in  2:levb) {alpha.beta[1,b] <-  - sum(alpha.beta[2:leva, b])}

#PRIORS
mu0 ~ dnorm(0, 0.0001)
for(a in 2:leva) {alpha[a] ~ dnorm(0, 0.0001)}
for(b in 2:levb) {beta[b] ~ dnorm(0, 0.0001)}
for(a in 2:leva) {for(b in 2:levb){
    alpha.beta[a,b] ~ dnorm(0, 0.0001) }}
tau ~ dgamma(0.001, 0.001)
s <- 1/sqrt(tau)

#PAIRWISE COMPARISONS

for(i in 1:1)  {for(j in i+1:2) {ca[i,j] <- alpha[i]-alpha[j]}}
for(i in 1:3)  {for(j in i+1:4) {cb[i,j] <- beta[i]-beta[j]}}

}
```

DATA 1

```
list(n =108, leva= 2, levb= 4)
```

→ DATA2 ←

INITS

```
list(mu0=0, alpha=c(NA, 0), beta=c(NA,0,0,0),
alpha.beta = structure(.Data=c(NA, NA, NA, NA, NA, 0,
0, 0),
.Dim=c(2,4)), tau = 1)
```

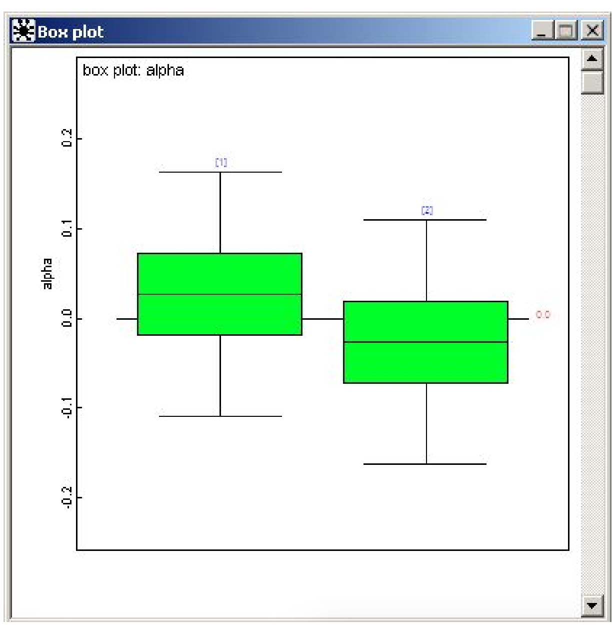
Node statistics obtained after burning 1000 samples:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha[1]	0.0267	0.06914	2.237E-4	-0.1089	0.02678	0.1622	1001	100000
alpha[2]	-0.0267	0.06914	2.237E-4	-0.1622	-0.02677	0.1089	1001	100000
alpha.beta[1,1]	-0.02636	0.1125	2.514E-4	-0.2467	-0.02591	0.1945	1001	100000
alpha.beta[1,2]	-0.1064	0.1239	5.496E-4	-0.3505	-0.1061	0.1366	1001	100000
alpha.beta[1,3]	0.2089	0.1237	5.429E-4	-0.03463	0.2091	0.4518	1001	100000
alpha.beta[1,4]	-0.07618	0.1171	5.152E-4	-0.3071	-0.07614	0.153	1001	100000
alpha.beta[2,1]	0.02636	0.1125	2.514E-4	-0.1945	0.02592	0.2468	1001	100000
alpha.beta[2,2]	0.1064	0.1239	5.496E-4	-0.1365	0.1061	0.3505	1001	100000
alpha.beta[2,3]	-0.2089	0.1237	5.429E-4	-0.4517	-0.2091	0.03465	1001	100000
alpha.beta[2,4]	0.07618	0.1171	5.152E-4	-0.1529	0.07614	0.3072	1001	100000
beta[1]	0.2033	0.1125	2.429E-4	-0.01638	0.2029	0.4251	1001	100000
beta[2]	0.2228	0.1242	5.517E-4	-0.02037	0.223	0.4675	1001	100000
beta[3]	0.05815	0.1236	5.534E-4	-0.1835	0.05783	0.3027	1001	100000
beta[4]	-0.4842	0.1176	4.902E-4	-0.7142	-0.4842	-0.2525	1001	100000
ca[1,2]	0.05339	0.1383	4.474E-4	-0.2178	0.05355	0.3245	1001	100000
cb[1,2]	-0.01952	0.1925	6.47E-4	-0.3967	-0.01972	0.3565	1001	100000
cb[1,3]	0.1451	0.192	6.438E-4	-0.2336	0.1451	0.5206	1001	100000
cb[1,4]	0.6874	0.1842	5.593E-4	0.3266	0.6863	1.051	1001	100000
cb[2,3]	0.1646	0.2057	9.673E-4	-0.2403	0.1652	0.5675	1001	100000
cb[2,4]	0.707	0.1987	8.794E-4	0.3165	0.7072	1.097	1001	100000
cb[3,4]	0.5423	0.1977	8.859E-4	0.154	0.542	0.9326	1001	100000
deviance	231.7	4.469	0.0183	225.1	231.0	242.3	1001	100000
s	0.709	0.05099	1.692E-4	0.6178	0.706	0.8179	1001	100000
tau	2.02	0.2869	9.287E-4	1.495	2.006	2.62	1001	100000

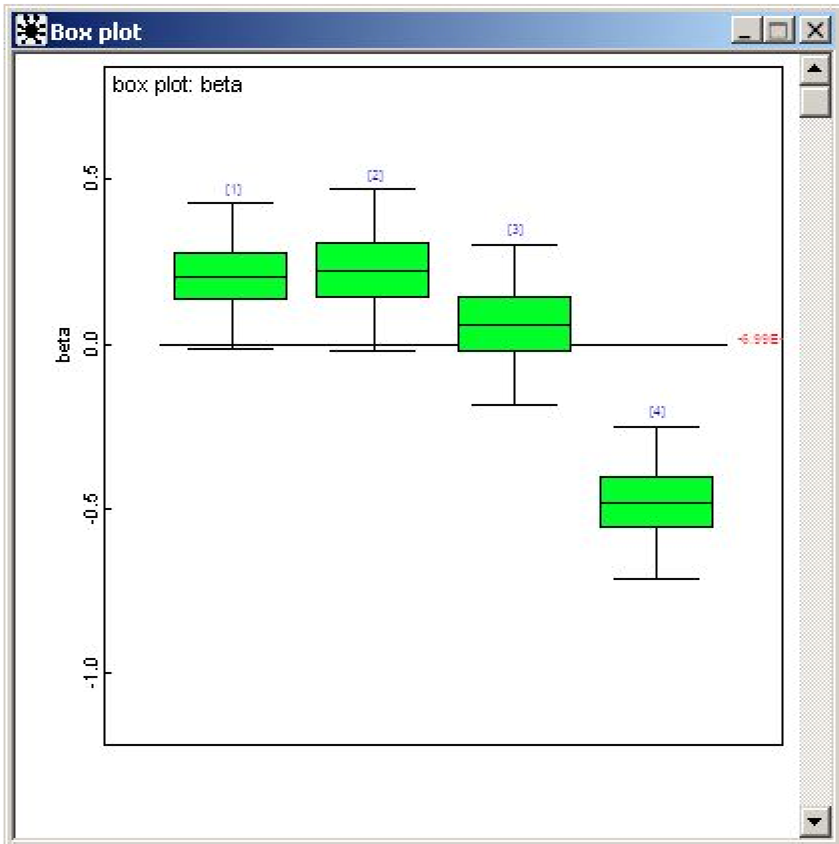
Since 0 is contained in the 95% CS: ca[1,2]; cb[1,2]; cb[1,3]; cb[2,3]; they are not different.

Similarly, we can look above in all the 95% CS for all the comparisons and can infer. If 0 is contained in the credible set, they are not different; Otherwise, if 0 is not contained in 95% CS, they are different.

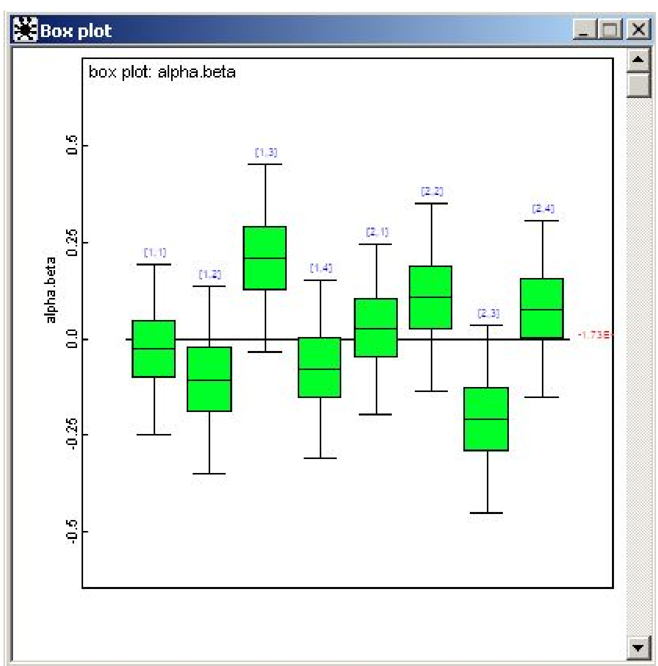
Boxplot for alpha:



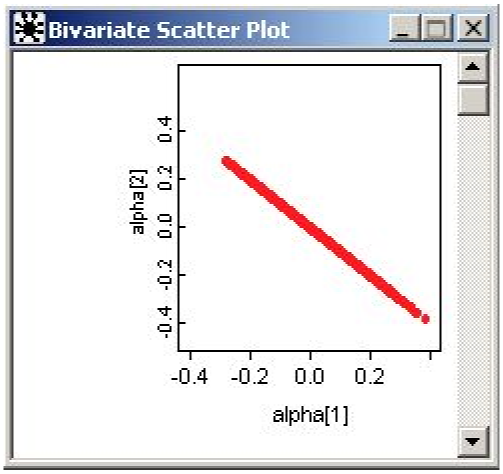
Boxplot for beta:



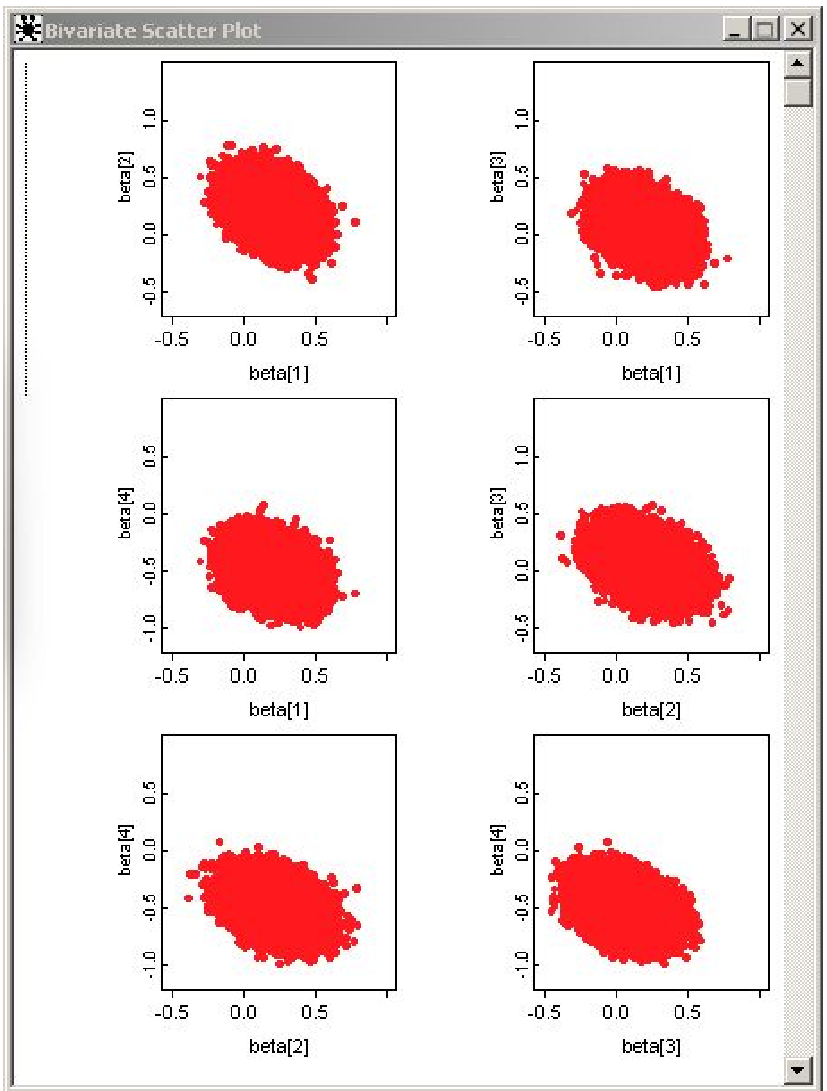
Boxplot for alpha.beta:



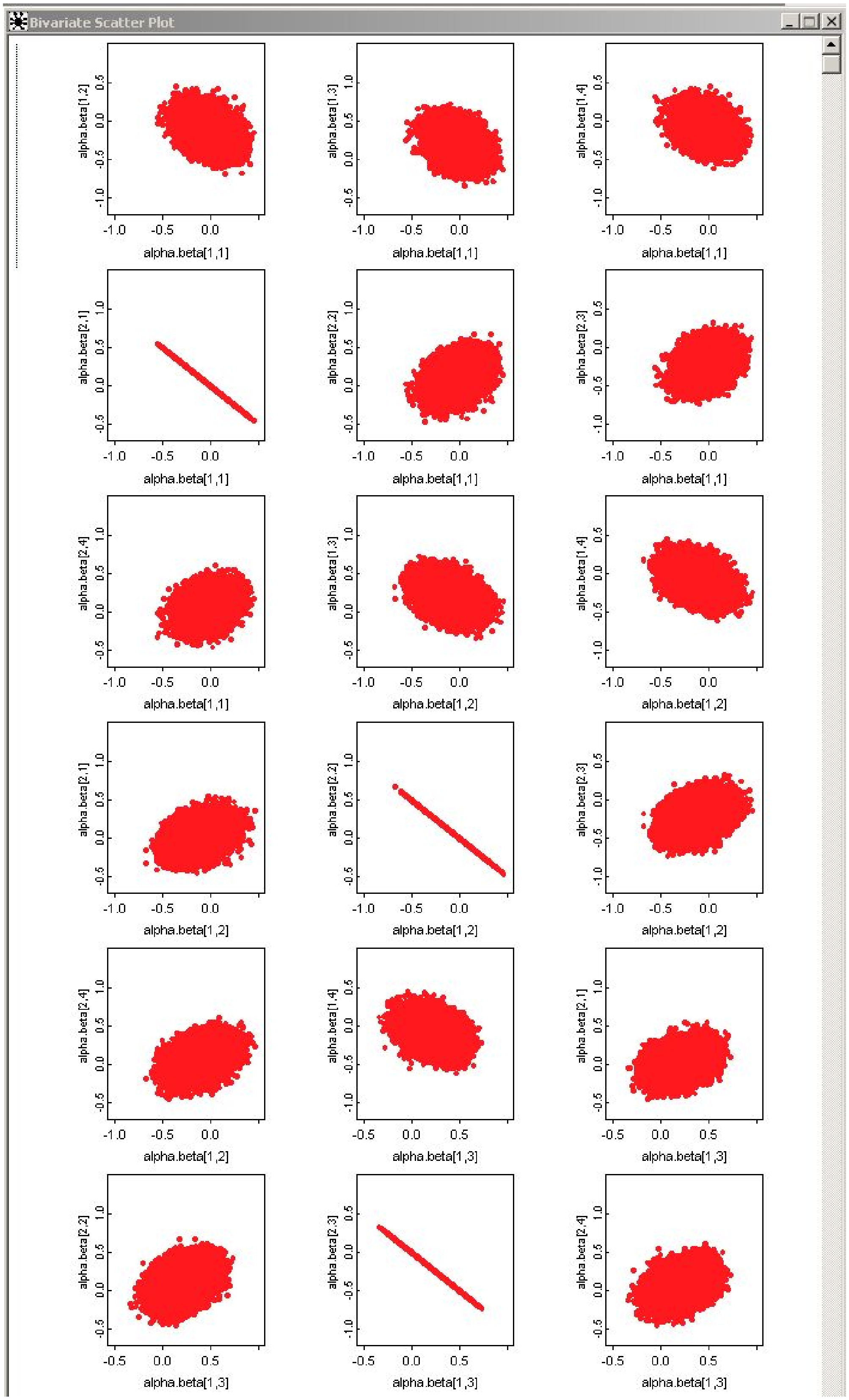
Correlation alpha:

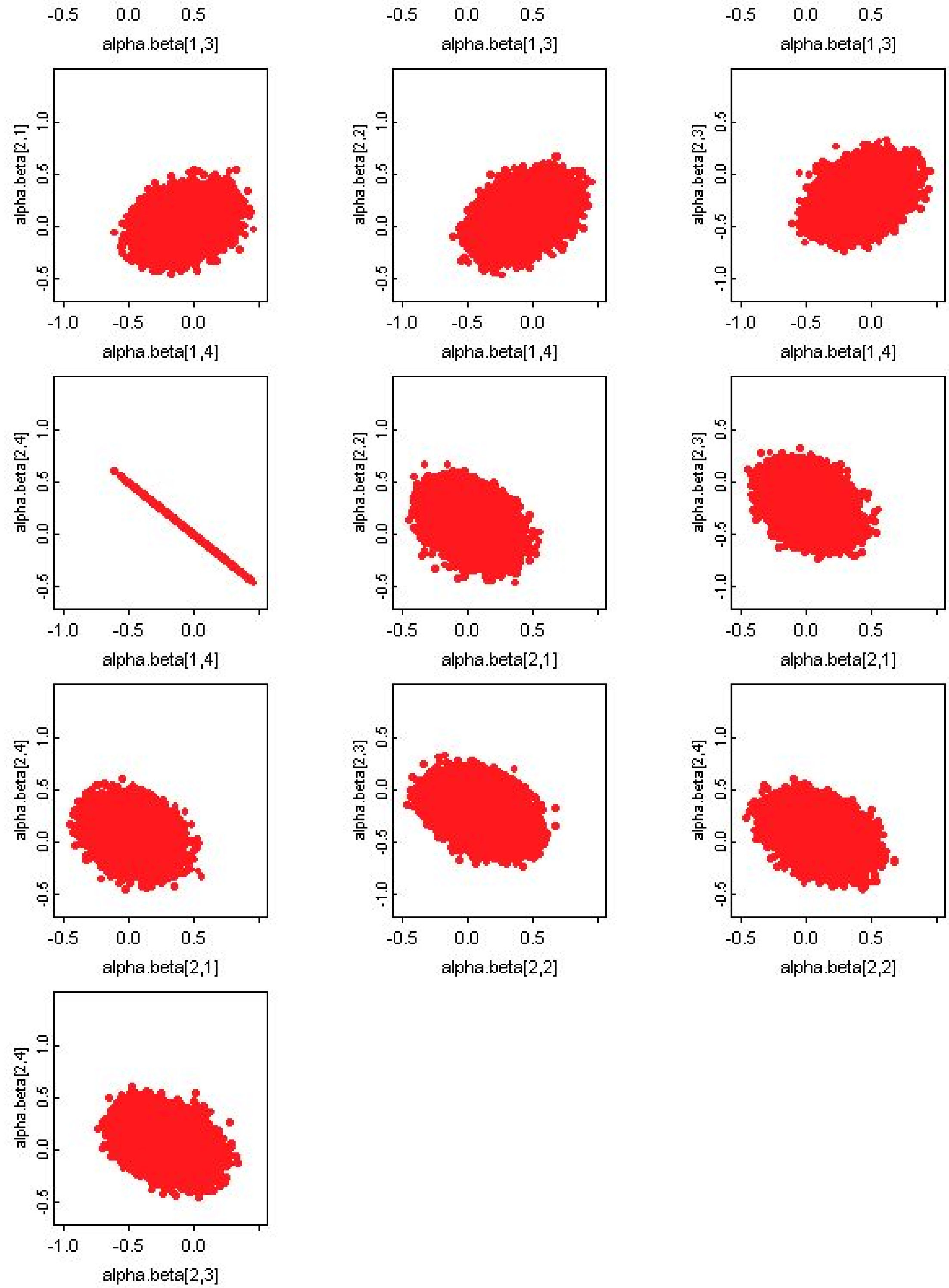


Correlation beta:

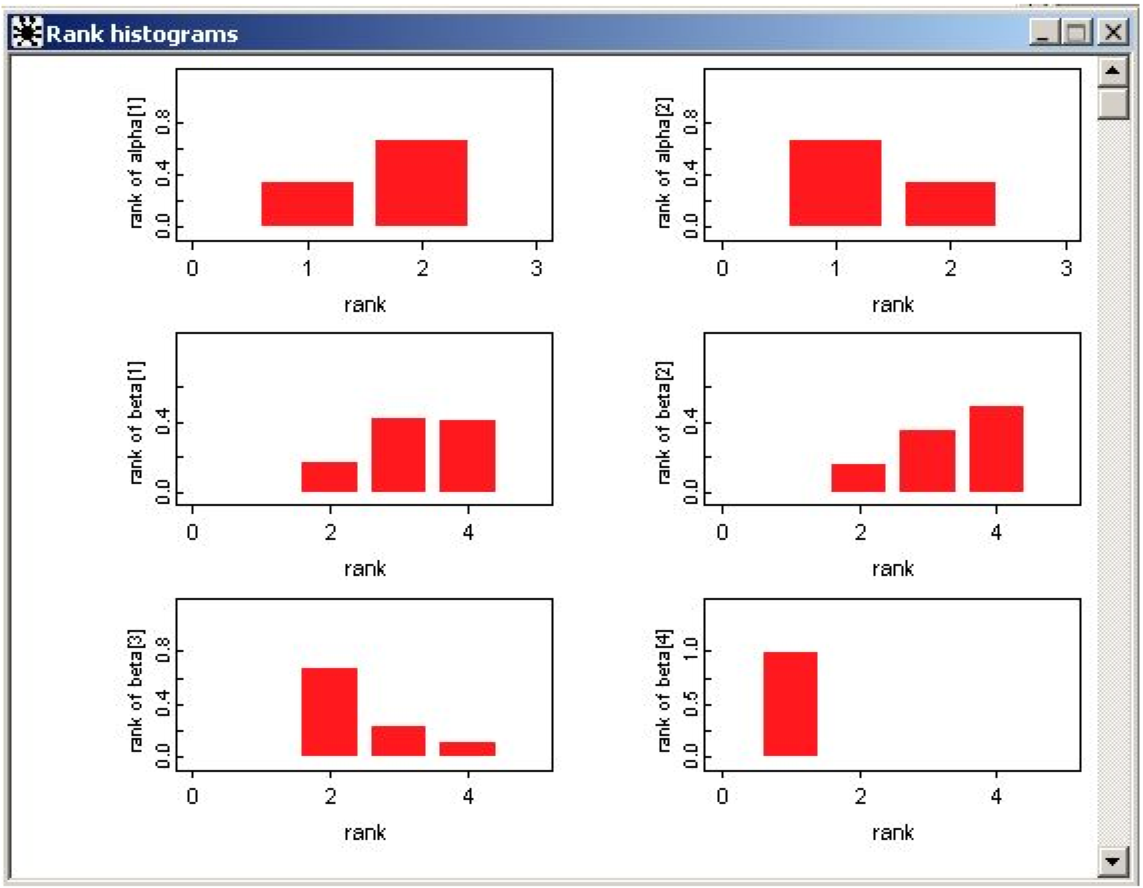


Correlation alpha.beta:




alpha.beta[1,3]
alpha.beta[2,1]
alpha.beta[2,2]
alpha.beta[2,3]
alpha.beta[1,4]
alpha.beta[2,4]
alpha.beta[2,1]
alpha.beta[2,2]
alpha.beta[2,3]

Rank:


Rank histograms
rank of alpha[1]
0.0 0.4 0.8
0 1 2 3
rank
rank of alpha[2]
0.0 0.4 0.8
0 1 2 3
rank
rank of beta[1]
0.0 0.4
0 2 4
rank
rank of beta[2]
0.0 0.4
0 2 4
rank
rank of beta[3]
0.0 0.4 0.8
0 2 4
rank
rank of beta[4]
0.0 0.5 1.0
0 2 4
rank

Question 3:

(a)

Used the below OpenBUGS code:

lumber0

```
model{
for( i in 1:N ) {
customers[i] ~ dpois(mu[i])
hunits0[i] <- hunits[i]/1000
aveinc0[i] <- aveinc[i]/10000
mu[i] <- exp( b0 + b1* hunits0[i] + b2* aveinc0[i] + b3* avehage[i] + b4* distcomp[i] + b5* diststore[i])
}

b0 ~ dnorm(0, 0.01)
b1 ~ dnorm(0, 0.01)
b2 ~ dnorm(0, 0.01)
b3 ~ dnorm(0, 0.01)
b4 ~ dnorm(0, 0.01)
b5 ~ dnorm(0, 0.01)


muystar <- exp( b0  + b1*0.72 + b2*7 + b3*6 + b4* 4.1 + b5*8)
ystar ~ dpois(muystar)
}

DATA
list(N=110)

➔| MILLER LUMBER COMPANY DATA 🡐

INITS
list(b0=0, b1=0, b2=0, b3=0, b4=0, b5=0)
```

Note here that, I rescaled the hunits by dividing it by 1000 and aveinc by dividing it by 10000 because of the numerical overflows that I was seeing in my OpenBUGS code. Therefore, all the analysis has been on coefficient b1 which is corresponding to hunits/1000 and b2 which is corresponding to aveinc/10000.

Got the below node statistics after burning 1000 samples:

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b0	2.783	0.218	0.02166	2.137	2.81	3.106	1001	10000
b1	0.6142	0.1536	0.0135	0.2934	0.6142	0.9139	1001	10000
b2	-0.111	0.0243	0.002286	-0.1571	-0.1114	-0.05824	1001	10000
b3	-0.003536	0.001826	1.278E-4	-0.007343	-0.003539	6.83E-5	1001	10000
b4	0.1816	0.02677	0.002516	0.1313	0.1811	0.236	1001	10000
b5	-0.1176	0.01619	0.001517	-0.1462	-0.1177	-0.07999	1001	10000
deviance	566.1	4.507	0.3751	560.4	565.0	576.9	1001	10000
muystar	9.328	0.6946	0.05218	8.046	9.293	10.81	1001	10000
ystar	9.323	3.092	0.05733	4.0	9.0	16.0	1001	10000

(b)

Based on the 95% credible set for each of the variables given below:

95% CS for b1 = [0.2934, 0.9139]

95% CS for b2 = [-0.1571, -0.05824]

95% CS for b3 = [-0.007343, 6.83E-5]

95% CS for b4 = [0.1313, 0.236]

95% CS for b5 = [-0.1462, -0.007999]

Since 0 lies in CS for b3, avehage can be ignored.

Also, if we round the upper limit of 95% CS for b2 and b5, they almost contain 0. Therefore, in my model, I would remove b2 and b5 too. I also cross validated this using Laud Ibrahim criteria and checking the deviance for each of the possible models by taking two covariates at a time and calculating and comparing each of the model's deviance. I found the model with covariates hunits and distcomp to have the lowest deviance.

Therefore, in my new model, I would only consider b1 and b4. The corresponding covariates are: hunits and distcomp.

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b1	0.2575	0.1094	0.009024	0.04789	0.2583	0.4615	1001	10000
b4	0.2611	0.01881	0.00148	0.2224	0.2619	0.299	1001	10000
deviance	643.3	2.582	0.17	640.6	642.6	650.0	1001	10000

As can be seen above, this model has increased deviance as compared to the model with all the predictors, but still, the deviance is much lower than the deviance of other possible models included other predictors by taking two covariates at a time.

Therefore, we can say that if we have to choose only 2 covariates, we can just choose hunits and distcomp.

(c)

The mean predicted response is given by ystar which is 9.323 and the average response is given by muystar 9.328.

The 95% for predicted number of customers in a representative 2-week period can be given by 95% CS of ystar which is [4, 16].