Problem 9.5:

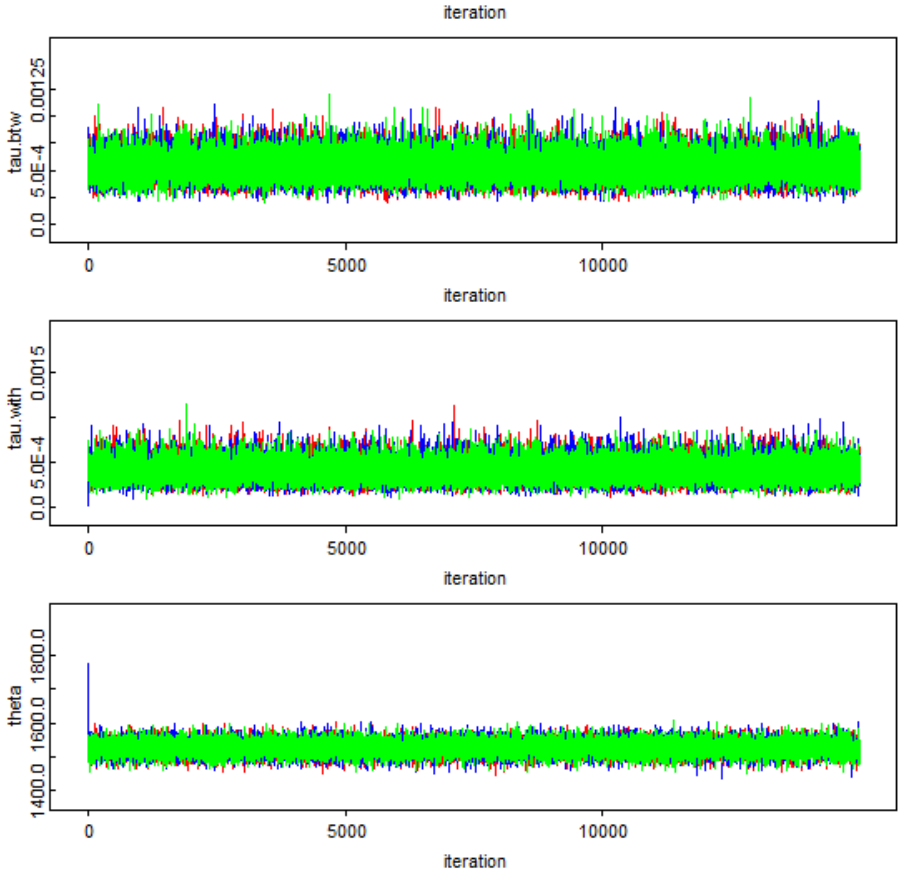
My OpenBugs have some problem on directed graph, I will draw the graph by hand on separate file.

1. All the $y_{ij}'s$ are not considered exchangeable because i and j are sampling from different loops in graph. The probability distribution on the $y_{1j}'s$ is different from that of $y_{2j}'s$.

2. All the $y_{2j}'s$ are considered exchangeable because they are all sampling from the same distribution with μ_2 and precision $\tau^2_{with.}$.

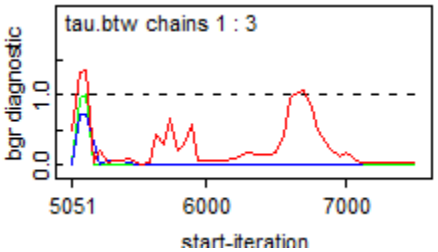
3. $\mu_i's$ are exchangeable because they have the same probability distributions which are drawn from the $Normal(\theta, \tau.btw)$.

4.



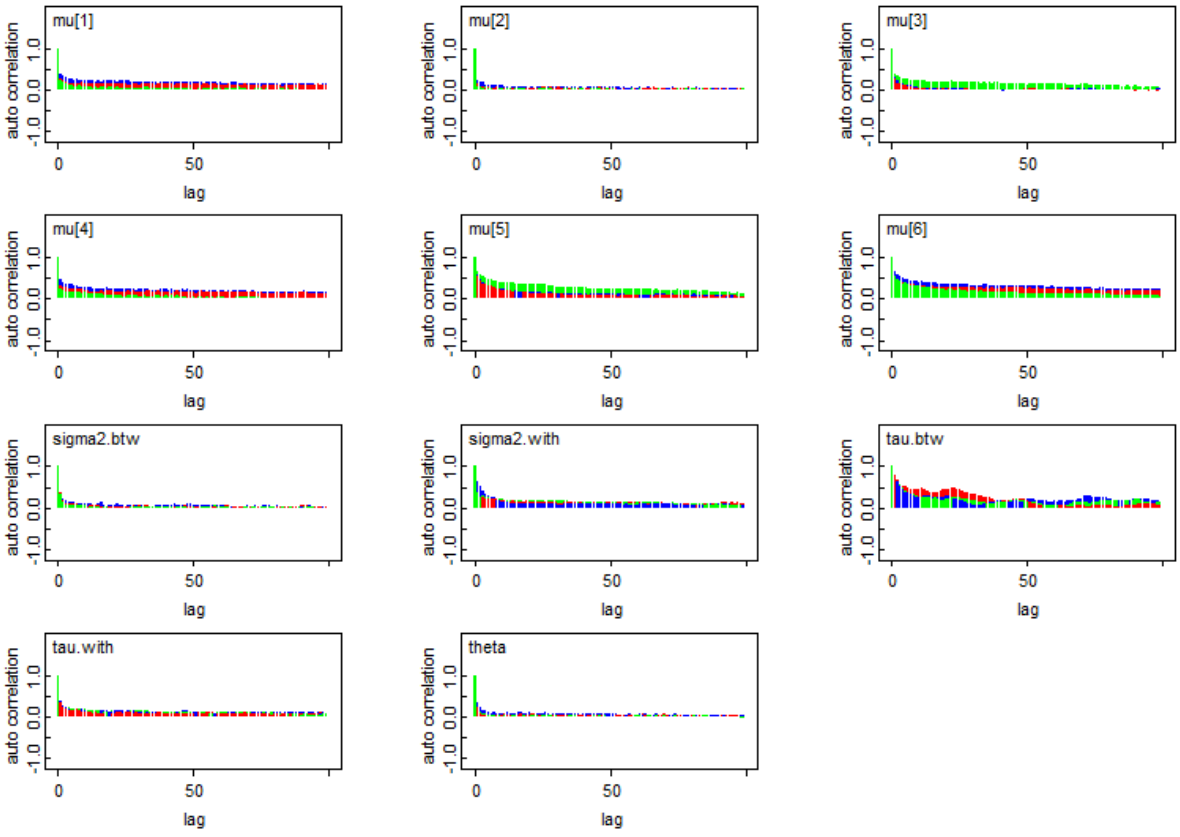
Both single chain and multi-chains show its convergence.

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu[1]	1510.0	21.03	0.1853	1469.0	1510.0	1551.0	1	15000
mu[2]	1528.0	21.02	0.1838	1487.0	1528.0	1569.0	1	15000
mu[3]	1556.0	21.04	0.1909	1514.0	1556.0	1597.0	1	15000
mu[4]	1504.0	21.15	0.1873	1464.0	1504.0	1545.0	1	15000
mu[5]	1584.0	21.42	0.1883	1542.0	1585.0	1625.0	1	15000
mu[6]	1483.0	21.54	0.1889	1441.0	1482.0	1525.0	1	15000
sigma2.btw	1974.0	464.2	4.355	1265.0	1911.0	3059.0	1	15000
sigma2.with	2675.0	1046.0	13.33	1484.0	2514.0	4697.0	1	15000
tau.btw	5.332E-4	1.193E-4	1.09E-6	3.271E-4	5.234E-4	7.908E-4	1	15000
tau.with	4.087E-4	1.18E-4	1.207E-6	2.129E-4	3.979E-4	6.739E-4	1	15000
theta	1527.0	20.46	0.1927	1487.0	1527.0	1567.0	1	15000



The bgr diagnostic of tau.btw doesn't seems good. The high values indicate that there might be extreme values.

5. After running another iterations autocorrelation plots, the outcome doesn't seem to be changed.



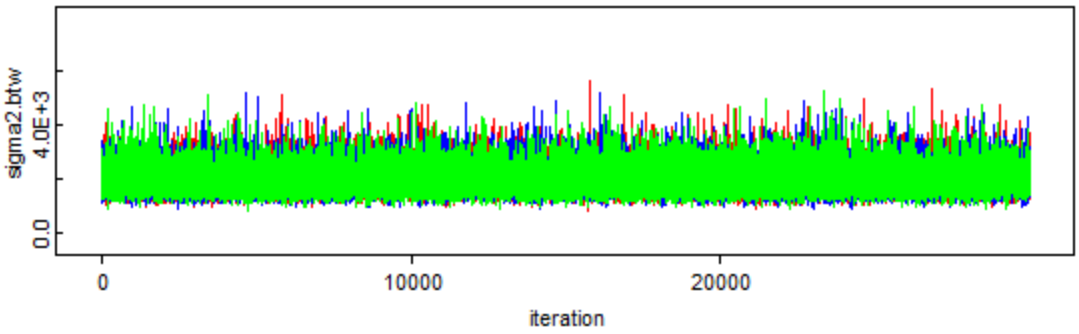
	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu[1]	1510.0	20.89	0.07577	1469.0	1510.0	1551.0	1	90000
mu[2]	1528.0	20.78	0.0734	1487.0	1528.0	1569.0	1	90000
mu[3]	1556.0	20.91	0.07899	1514.0	1556.0	1597.0	1	90000
mu[4]	1504.0	20.84	0.07531	1464.0	1504.0	1546.0	1	90000
mu[5]	1584.0	21.3	0.08023	1541.0	1585.0	1625.0	1	90000
mu[6]	1482.0	21.21	0.07577	1442.0	1482.0	1525.0	1	90000
sigma2.btw	1978.0	469.9	1.694	1261.0	1909.0	3084.0	1	90000
sigma2.with	2667.0	872.5	3.903	1498.0	2513.0	4725.0	1	90000
tau.btw	5.327E-4	1.202E-4	4.294E-7	3.243E-4	5.237E-4	7.93E-4	1	90000
tau.with	4.086E-4	1.169E-4	4.825E-7	2.116E-4	3.979E-4	6.675E-4	1	90000
theta	1527.0	20.42	0.08611	1487.0	1527.0	1568.0	1	90000

However, the MC error decreases for additional iterations.

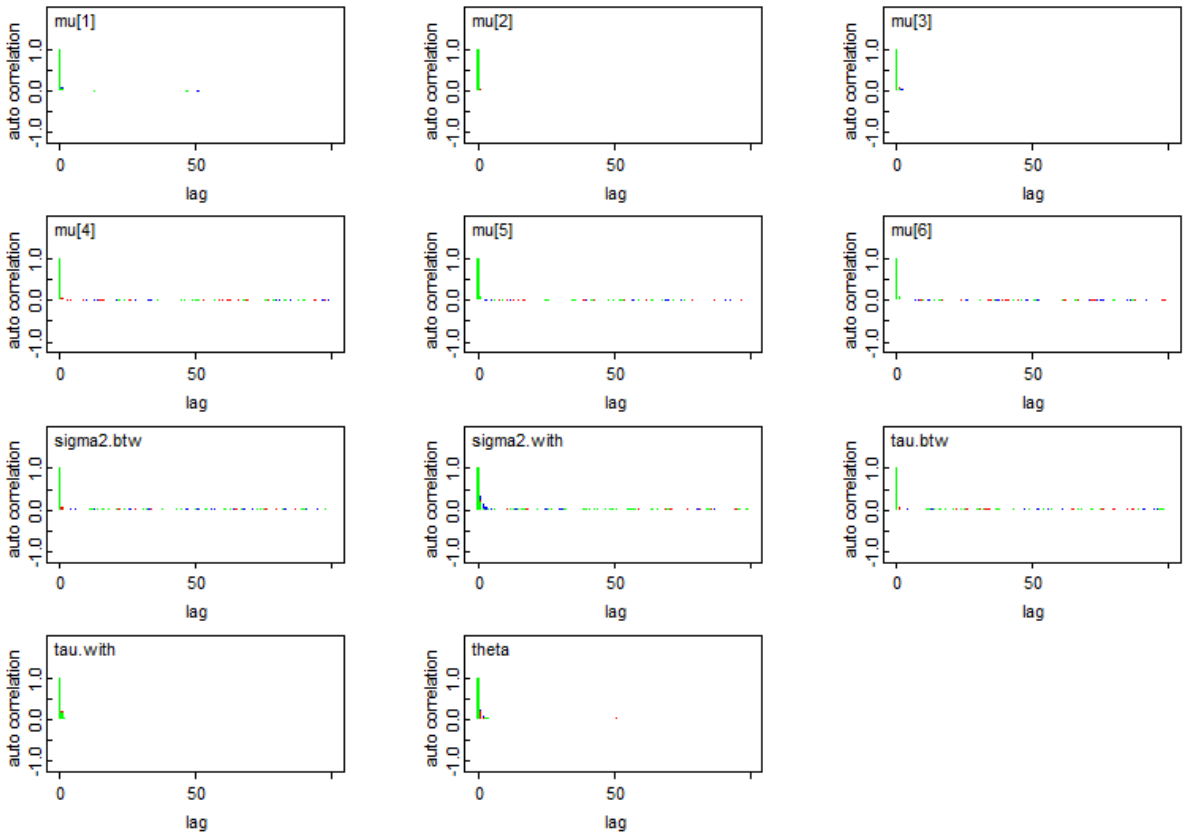
6. $\frac{\beta}{\alpha-1} = 2000, \frac{\beta^2}{(\alpha-1)^2(\alpha-2)} = 250000. \alpha = 18, \beta = 34000.$

7. Change tau.btw to be ~dgamma (18, 34000)

```
model
{
   for(i in 1 : batches) {
      mu[i] ~ dnorm(theta, tau.btw)
      for(j in 1 : samples) {
         y[i , j] ~ dnorm(mu[i], tau.with)
         cumulative.y[i , j] <- cumulative(y[i , j], y[i , j])
      }
   }
   sigma2.with <- 1 / tau.with
   sigma2.btw <- 1 / tau.btw
   tau.with ~ dgamma(0.001, 0.001)
   tau.btw ~ dgamma(18,34000)
   theta ~ dnorm(0.0, 1.0E-10)
}

#data
list(batches = 6, samples = 5,
   y = structure(
   .Data = c(1545, 1440, 1440, 1520, 1580,
    1540, 1555, 1490, 1560, 1495,
    1595, 1550, 1605, 1510, 1560,
    1445, 1440, 1595, 1465, 1545,
    1595, 1630, 1515, 1635, 1625,
    1520, 1455, 1450, 1480, 1445), .Dim = c(6, 5)))

#inits
list(theta=1500, tau.with=1, tau.btw=1)
list(theta=3000, tau.with=0.1, tau.btw=0.1)
list(mu = c(100, 300, 600,900, 1200, 1500), theta = 2000, tau.btw=0.0001, tau.with =10000)
```



There are no big changes for other plots. And for sigma2.btw, it seems to be converging, comparing to the previous one.

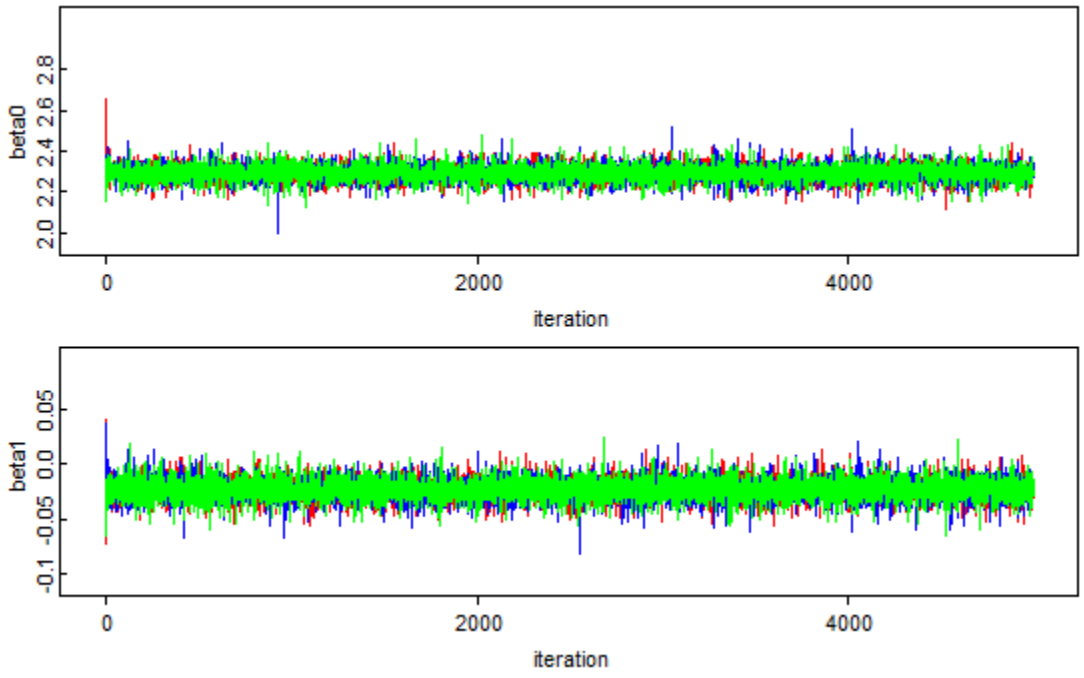


Both autocorrelation plots and bgr diagnostics show convergence of nodes.

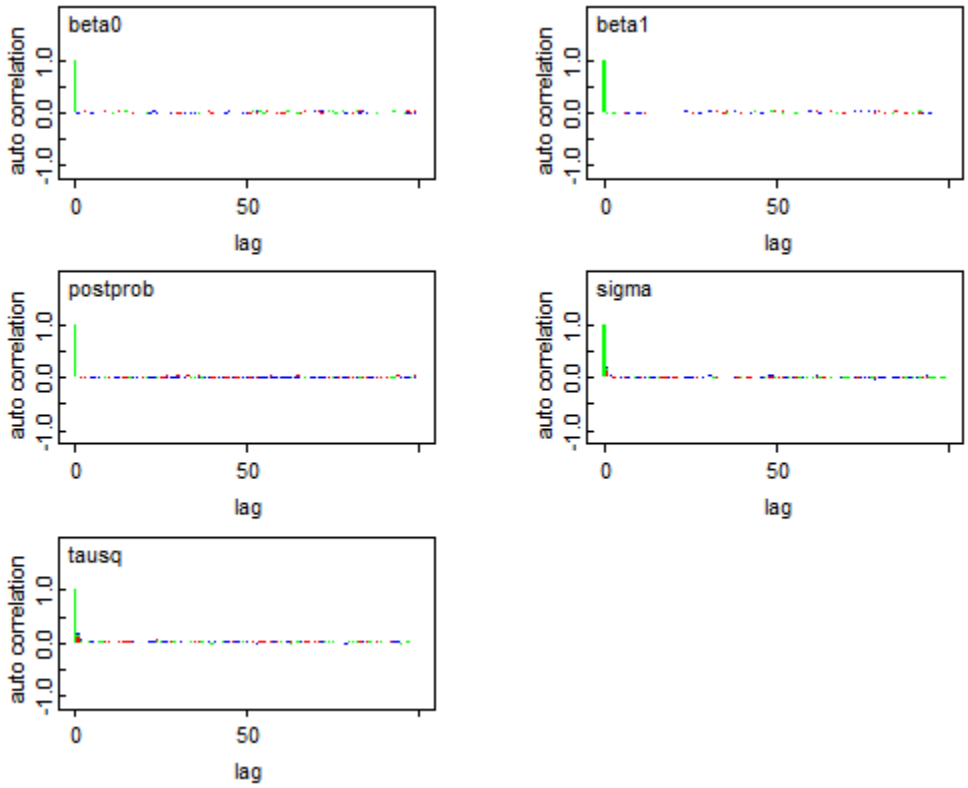
Problem 10.1:

```
model
{
for( i in 1:N) {
  xcent[i] <- x[i] - mean(x[])
}
for (i in 1:N) {
  mu[i] <- beta0 + beta1 * xcent[i]
  y[i] ~ dnorm( mu[i], tausq )
}
beta0 ~ dflat()
beta1 ~ dflat()
tausq ~ dgamma( 0.001, 0.001)
sigma <- 1 / sqrt(tausq) # regression standard deviation

postprob<-step(beta1)    #count the iterations in which beta1
}


#data
list( x = c(1997, 1998, 1999, 2000, 2001, 2002, 2003,
2004, 2005, 2006, 2007, 2008, 2009, 2010),
y = c(2.2952, 2.3435, 2.5512, 2.5531, 2.3918,
2.1546, 2.3596,
2.2431, 2.1725, 2.3162, 2.3504, 2.1926,
1.9638, 2.2025),
N = 14)

# inits
list( beta0 = 0, beta1 = 0, tausq = 1)
list( beta0 = 1, beta1 = 1, tausq = 2)
list( beta0 = 2, beta1 = 2, tausq = 3)
```

History:



Auto-cor:



Stats:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta0	2.292	0.03821	2.991E-4	2.216	2.292	2.367	1	15000
beta1	-0.02291	0.009532	8.051E-5	-0.04177	-0.02295	-0.004085	1	15000
postprob	0.01127	0.1055	8.615E-4	0.0	0.0	0.0	1	15000
sigma	0.1382	0.03134	2.986E-4	0.09238	0.1332	0.2132	1	15000
tausq	59.82	24.6	0.2332	22.02	56.38	117.2	1	15000

1. If we look at the autocorrelation plots, we can find the three chains are immediately converging.

2. Frequentists' outputs are:

```
> confint(brulelmout2)
                 2.5 %       97.5 %
(Intercept)  2.21749013  2.366820717
xcent       -0.04152854 -0.004484178
```

By comparison we find the mean of beta0 from OpenBugs is identical to the result in book, ignoring the rounded decimal places. Beta1 is quite similar but mean of beta1 from OpenBugs is a slightly larger than the result in the book.

3. Based on "stats", the mean of postprob is 0.0113, which is the posterior probability of H0 if we ran 5000 iterations. The p-value is 0.00955, which is smaller than Bayesians got. If we ran more iterations, p-value would decrease as the number of iterations increase. That's because of the random sampling variability in the MCMC output.

Bayesians' P(beta1>0|y) = 0.0113. Frequentists' P is 0.00955 for one-sided test.

Problem 10.2:

```
model
{
for( i in 1:N) {
  xcent[i] <- x[i] - mean(x[])
}
for (i in 1:N) {
  mu[i] <- beta0 + beta1 * xcent[i]
  y[i] ~ dnorm( mu[i], tausq )
}
beta0 ~ dnorm(2,16)
beta1 ~ dnorm(0,16)
tausq ~ dgamma( 0.001, 0.001)
sigma <- 1 / sqrt(tausq) # regression standard deviation

postprob<-step(beta1)    #count the iterations in which beta1
}


#data
list( x = c(1997, 1998, 1999, 2000, 2001, 2002, 2003,
2004, 2005, 2006, 2007, 2008, 2009, 2010),
y = c(2.2952, 2.3435, 2.5512, 2.5531, 2.3918,
2.1546, 2.3596,
2.2431, 2.1725, 2.3162, 2.3504, 2.1926,
1.9638, 2.2025),
N = 14)

# inits
list( beta0 = 1.5, beta1 = -0.5, tausq = 1)
list( beta0 = 2, beta1 = 0, tausq = 2)
list( beta0 = 2.5, beta1 = 0.5, tausq = 3)
```

The history, auto-cor, bgr all show pretty good convergence.

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta0	2.285	0.03773	2.967E-4	2.208	2.286	2.358	1	15000
beta1	-0.02288	0.00952	8.051E-5	-0.04172	-0.02292	-0.003961	1	15000
postprob	0.01127	0.1055	8.458E-4	0.0	0.0	0.0	1	15000
sigma	0.1382	0.03129	2.99E-4	0.09231	0.1333	0.2126	1	15000
tausq	59.78	24.6	0.2344	22.12	56.28	117.4	1	15000

The estimate of population intercept and slope don’t change. It makes sense because these two estimates in previous questions are exactly in the interval provided in this question. The mean posterior probability of H0 decrease just a little bit. It also makes sense since the previous prior is unif (0,1) and the coding shifts it.