10.9

10.9 Markov Chain Monte Carlo

Gibbs Sampler

Question 1

We should note that by the assumption that

$$\mathbf{X}(n) \xrightarrow{d} \mathbf{X} \sim \pi$$

This implies that

$$\lim_{n\to\infty} \mathbb{P}(\mathbf{X}(n) = \mathbf{x}) = \mathbb{P}(\mathbf{X} = \mathbf{x}) = \pi(\mathbf{x})$$

Note that for any $\mathbf{y} \in S$,

$$\begin{aligned}\mathbb{P}(\mathbf{X}(n+1) = \mathbf{y}) &= \sum_{x\in S} \mathbb{P}(\mathbf{X}(n+1) = \mathbf{y} \mid \mathbf{X}(n) = \mathbf{x})\mathbb{P}(\mathbf{X}(n) = \mathbf{x}) \\ &= \sum_{x\in S} \pi(\mathbf{x}, \mathbf{y})\mathbb{P}(\mathbf{X}(n) = \mathbf{x}).\end{aligned}$$

Here we use the law of total probability and that $\pi(\mathbf{x}, \mathbf{y}) = \mathbb{P}(\mathbf{X}(n+1) = \mathbf{y} \mid \mathbf{X}(n) = \mathbf{x})$. We can now take limits of both sides and as $|S| < \infty$ we can take the limit inside the sum without problem, therefore

$$\pi(\mathbf{y}) = \sum_{x\in S} \pi(\mathbf{x}, \mathbf{y})\pi(\mathbf{x}).$$

So π is an equilibrium distribution for this chain.

Football Data

Question 2

We are given the following prior distributions (and probability density functions)

$$\begin{aligned}
Y_{k,t}|\mu_k, \sigma_k^2 &\sim \mathrm{N}\big(\mu_k, \sigma_k^2\big) & f_{Y_{k,t}|\mu_k,\sigma_k^2}(y_{k,t}) &= (2\pi)^{-\frac{1}{2}}\big(\sigma_k^{-2}\big)^{\frac{1}{2}} e^{-\frac{1}{2\sigma_k^2}\left(y_{k,t}-\mu_k\right)^2} \\
\mu_k \ |\theta &\sim \mathrm{N}\big(\theta, \sigma_0^2\big) & f_{\mu_k|\theta}(\mu_k) &= \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(\mu_k-\theta)^2} \\
\sigma_k^{-2} &\sim \Gamma(\alpha_0, \beta_0) & f_{\sigma_k^{-2}}\big(\sigma_k^{-2}\big) &= \frac{\beta_0^{\alpha_0}\big(\sigma_k^{-2}\big)^{\alpha_0-1} e^{-\beta_0\sigma_k^{-2}}}{\Gamma(\alpha_0)} \\
\theta &\sim \mathrm{N}\big(\mu_0, \tau_0^2\big) & f_\theta(\theta) &= \frac{1}{\sqrt{2\pi\tau_0^2}} e^{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2}.
\end{aligned}$$

True for $k = 1, \ldots, K,\ t = 1, \ldots, T$, where $\sigma_0^2,\ \alpha_0,\ \beta_0,\ \mu_0,\ \tau_0^2$ are known constants. Now define the vectors $\boldsymbol{\sigma}^{-2},\ \mathbf{Y},\ \boldsymbol{\mu}$ such that $\boldsymbol{\sigma}_k^{-2} = \frac{1}{\sigma_k^2},\ \mathbf{Y}_{k,t} = Y_{k,t},\ \boldsymbol{\mu}_k = \mu_k$ for $k = 1, \ldots, K,\ t = 1, \ldots, T$. Note that

$$\mathbf{Y} \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \overset{d}{=} \mathbf{Y} \,|\, \boldsymbol{\mu}, \boldsymbol{\sigma}^{-2}$$

We can thus write the pdf for the joint distribution of $\mathbf{Y},\ \boldsymbol{\sigma}^{-2},\ \boldsymbol{\mu},\ \theta$ as follows

$$\begin{aligned}
f_{\mathbf{Y},\boldsymbol{\sigma}^{-2},\boldsymbol{\mu},\theta}\big(\mathbf{y},\ \boldsymbol{\sigma}^{-2},\ \boldsymbol{\mu},\ \theta\big) &= \left(\prod_{k=1}^{K}\left[\left(\prod_{t=1}^{T} f_{Y_{k,t}|\mu_k,\sigma_k^2}(y_{k,t})\right) f_{\mu_k|\theta}(\mu_k) f_{\sigma_k^{-2}}\big(\sigma_k^{-2}\big)\right]\right) f_\theta(\theta) \\
&= \left(\prod_{k=1}^{K}\left[(2\pi)^{-\frac{T}{2}}\big(\sigma_k^{-2}\big)^{\frac{T}{2}} \exp\left\{-\frac{1}{2}\big(\sigma_k^{-2}\big)\sum_{t=1}^{T}(y_{k,t}-\mu_k)^2\right\}\right.\right. \\
&\qquad \left.\left. \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(\mu_k-\theta)^2} \frac{\beta_0^{\alpha_0}\big(\sigma_k^{-2}\big)^{\alpha_0-1} e^{-\beta_0\sigma_k^{-2}}}{\Gamma(\alpha_0)}\right]\right) \frac{1}{\sqrt{2\pi\tau_0^2}} e^{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2} \\
&\propto \left(\prod_{k=1}^{K}\big(\sigma_k^{-2}\big)^{\frac{T}{2}+\alpha_0-1} \exp\left\{-\frac{1}{2}\big(\sigma_k^{-2}\big)\sum_{t=1}^{T}(y_{k,t}-\mu_k)^2\right.\right. \\
&\qquad \left.\left. -\frac{1}{2\sigma_0^2}(\mu_k-\theta)^2 - \beta_0\big(\sigma_k^{-2}\big) - 1\right\}\right) e^{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2}
\end{aligned}$$

From this we can read off all the terms that contribute to all possible posterior distributions, in the one dimensional case

$$\begin{aligned}
\pi\big(\mu_k \mid \boldsymbol{\mu}_{-k},\ \theta, \boldsymbol{\sigma}^2, \mathbf{y}\big) &\propto \exp\left\{-\frac{1}{2}\left[\big(T\sigma_k^{-2}+\sigma_0^{-2}\big)\mu_k^2 - 2\left(\sigma_k^{-2}\sum_{t=1}^{T} y_{k,t} + \sigma_0^{-2}\theta\right)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2\frac{1}{T\sigma_k^{-2}+\sigma_0^{-2}}}\left(\mu_k - \frac{\sigma_k^{-2}\sum_{t=1}^{T} y_{k,t} + \theta\sigma_0^{-2}}{T\sigma_k^{-2}+\sigma_0^{-2}}\right)^2\right\} \\
\pi\big(\theta \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}\big) &\propto \exp\left\{-\frac{1}{2}\left[\big(K\sigma_0^{-2}+\tau_0^{-2}\big)\theta^2 - 2\left(\sigma_0^{-2}\sum_{k=1}^{K}\mu_k + \mu_0\tau_0^{-2}\right)\theta\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2\frac{1}{K\sigma_0^{-2}+\tau_0^{-2}}}\left(\theta - \frac{\sigma_0^{-2}\sum_{k=1}^{K}\mu_k + \mu_0\tau_0^{-2}}{K\sigma_0^{-2}+\tau_0^{-2}}\right)^2\right\} \\
\pi\big(\sigma_k^{-2} \mid \boldsymbol{\sigma}_{-k}^2, \boldsymbol{\mu}, \theta, \mathbf{y}\big) &\propto \big(\sigma_k^{-2}\big)^{\frac{T}{2}+\alpha_0-1} \exp\left\{-\left(\beta_0 + \frac{1}{2}\sum_{t=1}^{T}(y_{k,t}-\mu_k)^2\right)\sigma_k^{-2}\right\}
\end{aligned}$$

Therefore

$$\begin{aligned}
\mu_k \ \ |\boldsymbol{\mu}_{-k},\ \theta, \boldsymbol{\sigma}^2, \mathbf{y} &\sim \mathrm{N}\left(\frac{\sigma_k^{-2}\sum_{t=1}^{T} y_{k,t} + \theta\sigma_0^{-2}}{T\sigma_k^{-2}+\sigma_0^{-2}}, \frac{1}{T\sigma_k^{-2}+\sigma_0^{-2}}\right) \\
\theta \quad |\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y} &\sim \mathrm{N}\left(\frac{\sigma_0^{-2}\sum_{k=1}^{K}\mu_k + \mu_0\tau_0^{-2}}{K\sigma_0^{-2}+\tau_0^{-2}}, \frac{1}{K\sigma_0^{-2}+\tau_0^{-2}}\right) \\
\sigma_k^{-2}|\boldsymbol{\sigma}_{-k}^2, \boldsymbol{\mu}, \theta, \mathbf{y} &\sim \Gamma\left(\alpha_0 + \frac{T}{2}, \beta_0 + \frac{1}{2}\sum_{t=1}^{T}(y_{k,t}-\mu_k)^2\right)
\end{aligned}$$

.

To find the marginal prior distribution of μ_k, consider the density function for the joint prior distribution of μ_k and θ

$$\begin{aligned}
f_{\mu_k,\theta}(\mu_k,\theta) &= f_{\mu_k|\theta}(\mu_k) f_\theta(\theta) \\
&= \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(\mu_k-\theta)^2} \frac{1}{\sqrt{2\pi\tau_0^2}} e^{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2} \\
&= \frac{1}{2\pi\sqrt{\sigma_0^2\tau_0^2}} \exp\left\{-\frac{1}{2}\left[(\sigma_0^{-2}+\tau_0^{-2})\theta^2 - 2(\mu_k\sigma_0^{-2}+\mu_0\tau_0^{-2})\theta + (\mu_k^2\sigma_0^{-2}+\mu_0^2\tau_0^{-2})\right]\right\} \\
&= \frac{\sqrt{2\pi\left(\sigma_0^{-2}+\tau_0^{-2}\right)^{-1}}}{2\pi\sqrt{\sigma_0^2\tau_0^2}} \exp\left\{-\frac{1}{2}\left[(\mu_k^2\sigma_0^{-2}+\mu_0^2\tau_0^{-2}) - \frac{\left(\mu_k\sigma_0^{-2}+\mu_0\tau_0^{-2}\right)^2}{\sigma_0^{-2}+\tau_0^{-2}}\right]\right\} \\
&\qquad \frac{1}{\sqrt{2\pi\left(\sigma_0^{-2}+\tau_0^{-2}\right)^{-1}}} \exp\left\{-\frac{1}{2\left(\sigma_0^{-2}+\tau_0^{-2}\right)^{-1}}\left(\theta - \frac{\mu_k\sigma_0^{-2}+\mu_0\tau_0^{-2}}{\sigma_0^{-2}+\tau_0^{-2}}\right)^2\right\} \\
&= \frac{1}{\sqrt{2\pi\left(\sigma_0^2+\tau_0^2\right)}} \exp\left\{-\frac{1}{2\left(\sigma_0^2+\tau_0^2\right)}(\mu_k-\mu_0)^2\right\} f_{\theta|\mu_k}(\theta) \\
&= f_{\mu_k}(\mu_k) f_{\theta|\mu_k}(\theta)
\end{aligned}$$

where

$$\begin{aligned}
f_{\theta|\mu_k}(\theta) &= \frac{1}{\sqrt{2\pi\left(\sigma_0^{-2}+\tau_0^{-2}\right)^{-1}}} \exp\left\{-\frac{1}{2\left(\sigma_0^{-2}+\tau_0^{-2}\right)^{-1}}\left(\theta - \frac{\mu_k\sigma_0^{-2}+\mu_0\tau_0^{-2}}{\sigma_0^{-2}+\tau_0^{-2}}\right)^2\right\} \\
f_{\mu_k}(\mu_k) &= \frac{1}{\sqrt{2\pi\left(\sigma_0^2+\tau_0^2\right)}} \exp\left\{-\frac{1}{2\left(\sigma_0^2+\tau_0^2\right)}(\mu_k-\mu_0)^2\right\}.
\end{aligned}$$

So we have now found the probability density functions for the prior distributions of $\theta \mid \mu_k$ and μ_k, and can see that both are of the form of a normal distribution and thus deduce that μ_k has prior marginal distribution

$$\mu_k \sim \mathrm{N}\left(\mu_0, \sigma_0^2+\tau_0^2\right).$$

Question 3

The code for the Gibbs Sampler can be found in the programs section

The following can be said for the assumptions made

- $\sigma_0 = 10$

  σ_0 can be seen as how mean score varies from team to team, this assumption implies that average score of approximately 99.7% of teams is within 30 points of the mean, this seams reasonable as when we look at the data, we see most of the data lies in the range 30 to 90.

- $\alpha_0 = 10^{-5}$, $\beta_0 = 10^{-3}$

  Allowing both α_0 and β_0 to be small suggests that we know very little about the distribution of σ_k prior to the process and thus allow mainly the data provided to determine the posterior distribution σ_k.

- $\mu_0 = 60$

  μ_0 can be seen as the overall average score of all teams combined, so we have assumed that the average over all teams is 60

- $\tau_0 = 20$

  τ_0 can be seen as the variance of the average score, so the above assumes that the average score over all teams could vary a lot, i.e we do not know much about their performance (as $\approx 99.7\%$ chance of lying within 3 standard deviations of the mean for a normal distribution, this along with the assumption $\mu_0 = 60$ says the mean score likely to be anywhere from 0 to 120, and as the minimum score is 0 and the maximum score is 114, this is a safe assumption)

We can initialise the Gibbs Sampler with naive approximations for all the variables i.e

$$\begin{aligned}
\mu_k &\approx \frac{1}{T}\sum_{t=1}^{T} Y_{k,t} \\
\sigma_k^2 &\approx \frac{1}{T}\sum_{t=1}^{T} Y_{k,t}^2 - \left(\frac{1}{T}\sum_{t=1}^{T} Y_{k,t}\right)^2 \\
\theta &\approx \frac{1}{KT}\sum_{t=1}^{T}\sum_{k=1}^{K} Y_{k,t}
\end{aligned}$$

This choice of starting values should not matter in the long run as we expect the sampler to tend in distribution to the invariant distribution of all the parameters.

Question 4

For our approximation of $\mathbb{E}[f(\mathbf{X})]$ we make use of the fact

$$\sum_{n=1}^{N} f(\mathbf{x}^n) \xrightarrow{N\to\infty} \mathbb{E}[f(\mathbf{X})]$$

so

$$\mathbb{E}[f(\mathbf{X})] \approx \sum_{n=1}^{N} f(\mathbf{x}^n) \tag{1}$$

for N sufficiently large, where $\mathbf{x}^n$ is the n^{th} iteration of the Gibbs sampler, and N is the total number of iterations. Let $\mathbf{X} = \boldsymbol{\mu}, \boldsymbol{\sigma}, \theta \mid \mathbf{y}$, then $\mathbf{x}^n$ represents the n^{th} iteration of the Gibbs sampler programmed above, we will therefore denote

$$\mathbf{x}^n = \left(\hat{\mu}_{1,n}, \ldots, \hat{\mu}_{K,n}, \hat{\sigma^2}_{1,n}, \ldots, \hat{\sigma^2}_{K,n}, \hat{\theta}_n\right) = \left(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\sigma}^2}_n, \hat{\theta}_n\right),$$

then by taking $f(\mathbf{x}) = \mathbf{x}$ and looking at each entry, we get

$$\begin{aligned}
\mathbb{E}[\mu_k \mid \mathbf{y}] &\approx \sum_{n=1}^{N} \hat{\mu}_{k,n} \\
\mathbb{E}\left[\sigma_k^2 \mid \mathbf{y}\right] &\approx \sum_{n=1}^{N} \hat{\sigma^2}_{k,n} \\
\mathbb{E}[\theta \mid \mathbf{y}] &\approx \sum_{n=1}^{N} \hat{\theta}_n
\end{aligned}$$

i.e. approximations of the posterior mean for each parameter. Now if we consider $f(\mathbf{x}) = \mathbb{1}_{\{a\le\theta\le b\}}$, then

$$\mathbb{P}(a \le \theta \le b \mid \mathbf{y}) \approx \sum_{n=1}^{N} \mathbb{1}_{\left\{a\le\hat{\theta}_n\le b\right\}}$$

So we can choose a partition $a_0 < a_1 < \ldots < a_m$ such that $a_0 < \min\hat{\theta}_n$, $a_m > \max\hat{\theta}_n$ which is equispaced, we can then create a histogram using the above approximation and note that the probability that θ lies on the boundary of two boxes is 0 so we can use the above approximation for

$\mathbb{P}(a < \theta \leq b \mid \mathbf{y})$ and $\mathbb{P}(a \leq \theta < b \mid \mathbf{y})$. The R function `hist` automatically does this with each partitions representing $\mathbb{P}(a_i < \theta \leq a_{i+1} \mid \mathbf{y})$ for $i > 0$ and $\mathbb{P}(a_0 \leq \theta \leq a_1 \mid \mathbf{y})$, when given the values of $\theta \mid \mathbf{y}$ and set `freq = false`.

All of the posterior means were stored in a file, the output of which is shown below

```
The posterior mean of Arsnal mean : 80.5388899134502
The posterior mean of Asten Villa mean : 51.8994369066947
The posterior mean of Blackborn Rovers mean : 50.4212988615273
The posterior mean of Boltin Wandrers mean : 51.6045368589237
The posterior mean of Charlston Athletic mean : 47.6933970650255
The posterior mean of Chelsea Buns mean : 68.0949126223017
The posterior mean of Evraton mean : 50.7358689936067
The posterior mean of Fullem mean : 48.5484113709561
The posterior mean of Livurpule mean : 61.2885855165455
The posterior mean of Manchester Ununited mean : 74.5365946496394
The posterior mean of Middlesbro mean : 49.8048703800766
The posterior mean of Newcassel Divided mean : 59.2698614733449
The posterior mean of Slothampton mean : 47.7995497755777
The posterior mean of Tottenham Coldspur mean : 51.4745690071147
The posterior mean of Arsnal variances : 96.6486036526048
The posterior mean of Asten Villa variances : 68.5221501354221
The posterior mean of Blackborn Rovers variances : 115.207399929893
The posterior mean of Boltin Wandrers variances : 130.955031219654
The posterior mean of Charlston Athletic variances : 24.9227221187762
The posterior mean of Chelsea Buns variances : 367.990299948751
The posterior mean of Evraton variances : 164.640972888534
The posterior mean of Fullem variances : 34.6062399159848
The posterior mean of Livurpule variances : 178.788672992579
The posterior mean of Manchester Ununited variances : 56.4543491652934
The posterior mean of Middlesbro variances : 25.3131582278894
The posterior mean of Newcassel Divided variances : 211.301550118149
The posterior mean of Slothampton variances : 142.998177502306
The posterior mean of Tottenham Coldspur variances : 47.4141341208647
The posterior mean of theta : 54.6717646511728
```

And the graph for the distribution of $\theta \mid \mathbf{y}$ is saved in a pdf, the result is shown in Figure 1

Question 5

To approximate $\mathbb{P}(\mu_k > \theta \mid \mathbf{y})$ we can again use Equation 1 and the fact $\mathbb{P}(\mu_k > \theta \mid \mathbf{y}) = \mathbb{E}\big[\mathbb{1}_{\{\mu_k > \theta\}} \mid \mathbf{y}\big]$ so

$$\mathbb{P}(\mu_k > \theta \mid \mathbf{y}) \approx \sum_{n=1}^{N} \mathbb{1}_{\{\hat{\mu}_{k,n} > \hat{\theta}_n\}}$$

I chose to do this for all teams and store the results in a text file, the output of which is shown below

```
Probability mean score of Arsnal is greater than average is: 0.996500349965003
Probability mean score of Asten Villa is greater than average is: 0.23977602239776
Probability mean score of Blackborn Rovers is greater than average is: 0.170882911708829
Probability mean score of Boltin Wandrers is greater than average is: 0.253874612538746
Probability mean score of Charlston Athletic is greater than average is: 0.0238976102389761
Probability mean score of Chelsea Buns is greater than average is: 0.956004399560044
Probability mean score of Evraton is greater than average is: 0.214078592140786
```

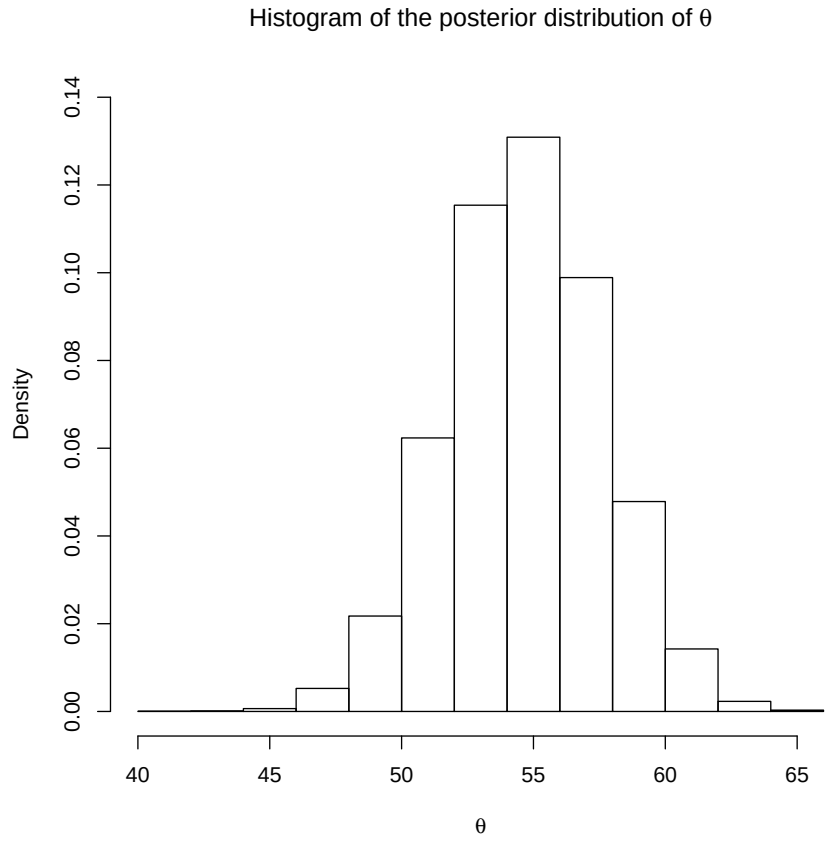


Figure 1: Histogram for the approximation of distribution of $\theta \mid \mathbf{y}$

```
Probability mean score of Fullem is greater than average is: 0.0475952404759524
Probability mean score of Livurpule is greater than average is: 0.895810418958104
Probability mean score of Manchester Ununited is greater than average is: 0.998200179982002
Probability mean score of Middlesbro is greater than average is: 0.0833916608339166
Probability mean score of Newcassel Divided is greater than average is: 0.811018898110189
Probability mean score of Slothampton is greater than average is: 0.0854914508549145
Probability mean score of Tottenham Coldspur is greater than average is: 0.198980101989801
```

Question 6

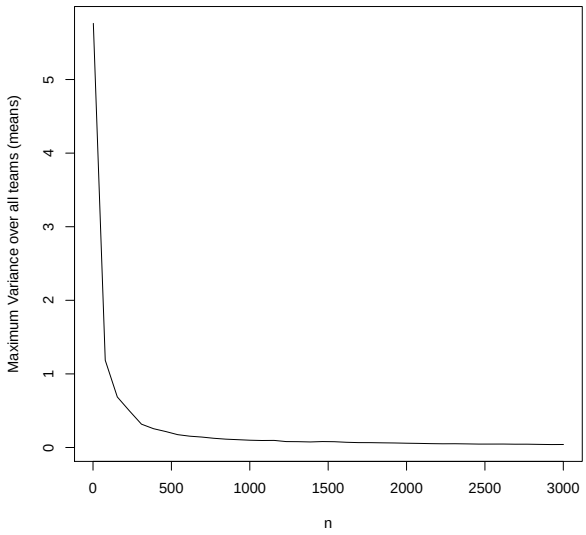
I wish to look at the overall performance of this algorithm for all teams so instead of considering the sample variance for each team individually after n iterations, I will instead consider the maximum sample variance of all the teams after n iterations. Doing this I get the graphs shown in Figure 2 and we see that the algorithm converges very quickly. To investigate the rate of convergence we will denote the maximum sample variance of the Gibbs sampler approximation of the mean after n iterations as $s^2_{\mu,n}$ and the maximum sample variance of the Gibbs sampler approximation of the probability a team is greater than average after n iterations $s^2_{\mathbb{P},n}$. We will plot $\log s^2_{\mu,n}$ against n and $\log s^2_{\mathbb{P},n}$ against n and look at the gradient, doing this we get the results shown in Figure 3 and this suggests that the gradient is either

1. converging to a constant value, $c < 0$

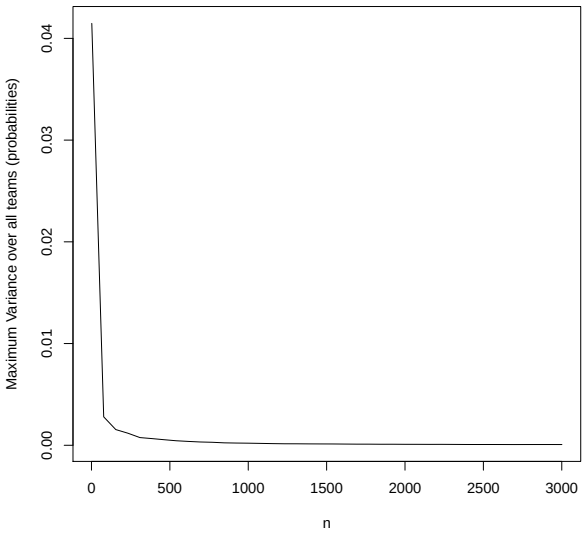
   This suggests that $\frac{s^2_{\mu,n}}{n^c} \xrightarrow{N\to\infty} d$ where d is some non-zero constant, so $s^2_{\mu,n} \sim dn^c$

2. converging to 0 from below

   This would suggest that at a certain point doing more iterations won't lead to much of an improvement to the performance of our estimation.



(a) Graph to show the maximum sample variance of Gibbs sampler approximation of the mean over all teams taken at regular intervals



(b) Graph to show the sample variance of Gibbs sampler approximation of the probability that a team is greater than average taken at regular intervals

Figure 2: Graphs showing the performance of Gibbs sampler to approximate the mean and probability a team is greater than average over all teams

Question 7

We have seen in the previous question that the algorithm seems to converge quickly to the values of $\mathbb{E}[\mu_k \mid \mathbf{y}]$ and $\mathbb{P}(\mu_k < \theta \mid \mathbf{y})$, which suggests that the algorithm may converge quickly to the invariant distribution of the Markov chain, so we expect that

$$((\mathbf{X}(m_0), \ldots, \mathbf{X}(m_0 + (n-1))) \overset{d}{\approx} (\mathbf{X}(m_1), \ldots, \mathbf{X}(m_1 + (n-1)))$$

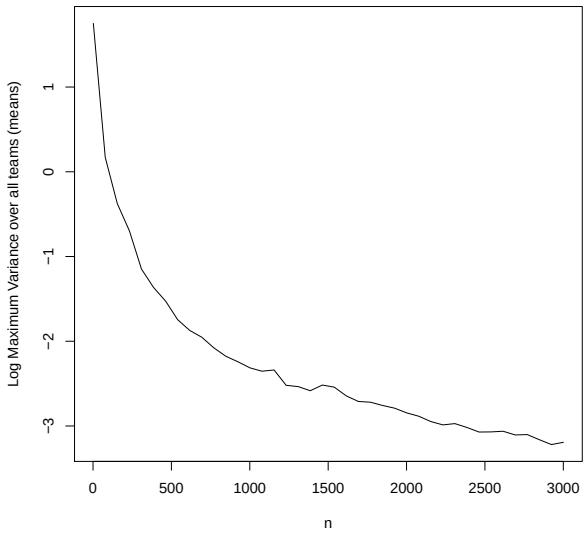
for $m_0, m_1, n \in \mathbb{N}$ where $\overset{d}{\approx}$ signifies approximate equality in distribution. So sample variances will be approximately independent of the number of pre-iterations we perform, which is what we see in Figure 4. This shows the number of pre-iterations we perform has little to no impact on the accuracy of our estimations, but that including more samples in our estimation improves the accuracy (which we saw already in question 6).

Question 8

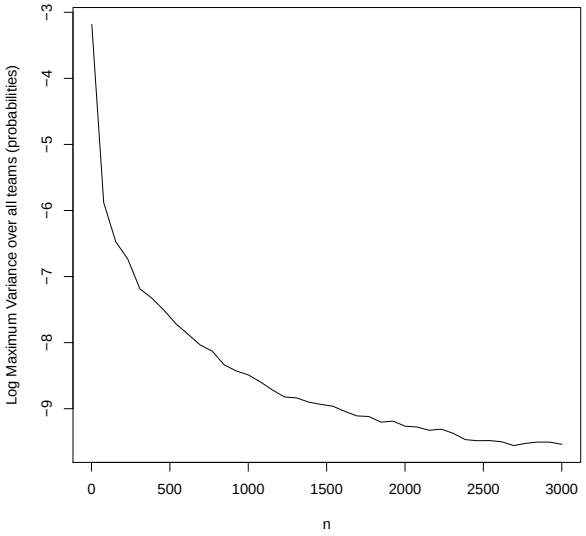
For this we will consider 200 different starting points with distributions decided by taking μ_k, and θ from uniform distributions with minimum 0 and maximum 120, and taking σ_k from uniform distributions with minimum 0.001 and maximum 40, and allow 200 iterations of the Gibbs Sampler.

We will first examine our estimates for the posterior distribution of θ using our new starting points compared to our original estimate. We can see how the new estimates performed in Figure 5, where we have only looked at the results using the first 5 starting points to reduce clutter. We can see that the distributions have similar shape, but still vary between starting points. If we look at the values of θ in the first few iterations

```
39.3425481487066 60.1627645463189 52.786276406036 57.8832853585804 50.1221830073268
65.8657679054886 52.1196909825922 59.9976649406668 54.8272253305758 57.2404282054413
93.4752504620701 57.4781414235093 59.0149651831041 54.3313634617734 47.8786514499538
```



(a) Graph to show the log of the maximum sample variance of Gibbs sampler approximation of the mean over all teams taken at regular intervals



(b) Graph to show the log of the sample variance of Gibbs sampler approximation of the probability that a team is greater than average taken at regular intervals

Figure 3: Graphs showing the rate of convergence of Gibbs sampler to approximate the mean and probability a team is greater than average over all teams

```
8.25819287449121 56.5117539448716 53.7147991890955 49.5080885973148 52.7950973056742
4.16899301111698 54.5215706749543 50.4630981893662 51.7633573312059 46.4246619136566
```

and the last few iterations

```
50.517211954027 50.1587216407981 55.927192496777 56.8629953796826 52.4446625381435
53.8896125602433 53.1526503698897 55.7890457597447 55.8792530100752 50.4962961960895
54.6648077999801 55.8993775335869 57.080893405556 49.9120555029468 57.2755504469562
50.5458757717354 55.368084066427 53.495679076741 52.4018395029388 51.6903568373098
56.0272708002103 55.4665021160861 56.2999136688187 51.5106889011464 52.0017902438653
```

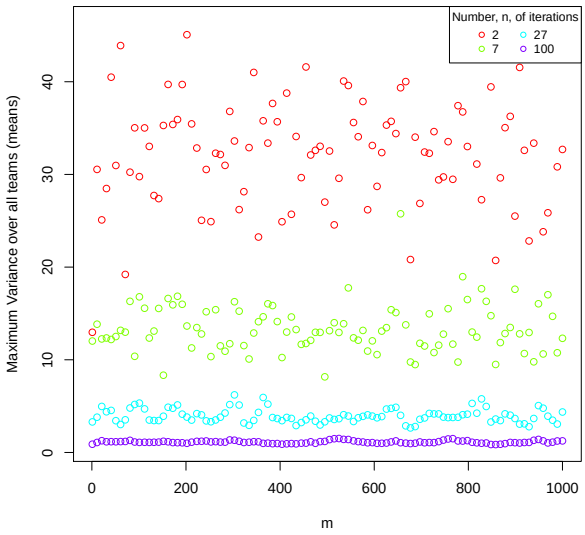
We see that after the first one or two iterations the values seems to have quickly converged to some equilibrium distribution, suggesting that performing a small number of pre-iterations may increase performance, but doing much more will not result in much improvement.

We will now examine our estimates for the posterior means of μ_k, σ_k, and our estimates of $\mathbb{P}(\mu_k < \theta \mid \mathbf{y})$. We can do this by simply looking at the standard deviation of our estimates and look at the max standard deviation over all estimates. We need to be careful in observing that μ_k, σ_k and $\mathbb{P}(\mu_k < \theta \mid \mathbf{y})$ are all of different magnitudes so the max standard deviation should be taken over all of them separately, doing this we get

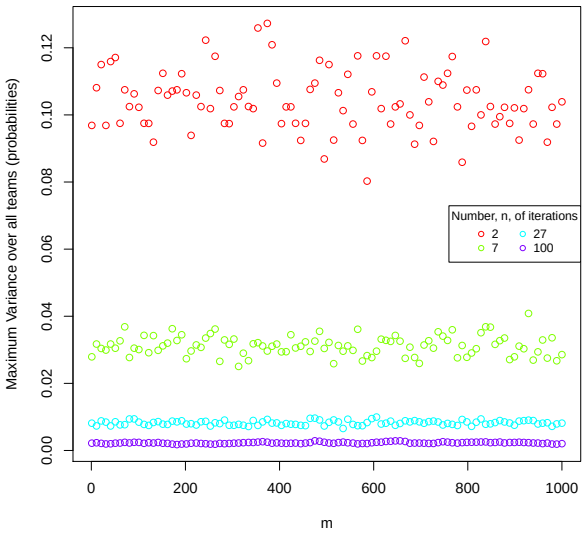
```
Maximum standard deviation for posterior mean of mu_k is 0.717597063469652
Maximum standard deviation for posterior mean of sigma_k^2 is 55.6439688303647
Maximum standard deviation for posterior mean of theta is 0.315121957142704
Maximum standard deviation for posterior probability mu_k > theta is 0.0331471652034635
```

This suggests that our estimation is good in all the scenarios in which we used it except possibly for estimating σ_k^2, for which we can look at variance alongside the expected value.

```
Expected value Standard Deviation
88.2934248982461 36.4951501629494
70.7214078556625 8.78178090232529
```



(a) Graph to show the maximum sample variance of Gibbs sampler approximation of the mean over all teams taken at regular intervals for $n = 2, 7, 27, 100$ iterations



(b) Graph to show the sample variance of Gibbs sampler approximation of the probability that a team is greater than average taken at regular intervals for $n = 2, 7, 27, 100$ iterations

Figure 4: Graphs showing the performance of Gibbs sampler to approximate the mean and probability a team is greater than average over all teams for $n = 2, 7, 27, 100$ iterations

```
110.721604168298 12.8462866155918
132.728681136333 15.6190001973469
25.6217429937799 3.51028388597347
365.195404854573 55.6439688303647
165.242770204858 17.5292812303321
34.4972228721523 4.90673800622719
181.774535109379 24.2224828073627
54.4954127838439 13.0055946519802
26.2974675326142 3.36741272403772
208.604818461295 21.5480054679443
143.496838747908 19.5200165679898
47.1232444784511 6.6539333898024
```

We see that the compared to the expectation, the standard deviation is relatively small, suggesting our estimates for σ_k^2 are good.

Now we can see that in the case where we randomly guess a starting point, the use of pre-iterations in some cases leads to a noticeable improvement in our estimation, but in other cases we do not see much of improvement, seen in Figure 6. This suggests that although pre-sampling may improve the accuracy of the estimation, it would be much more beneficial to simply use more samples in our estimation of the parameters.

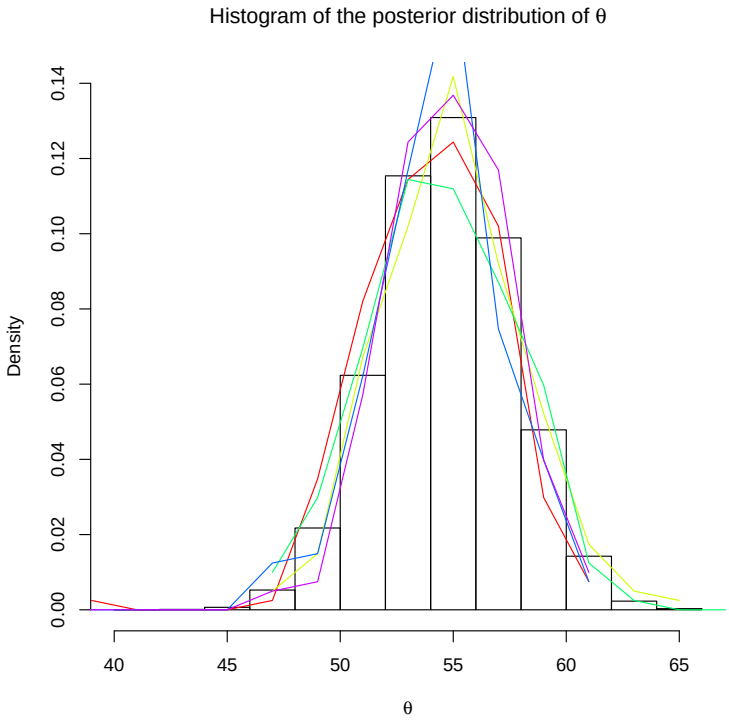
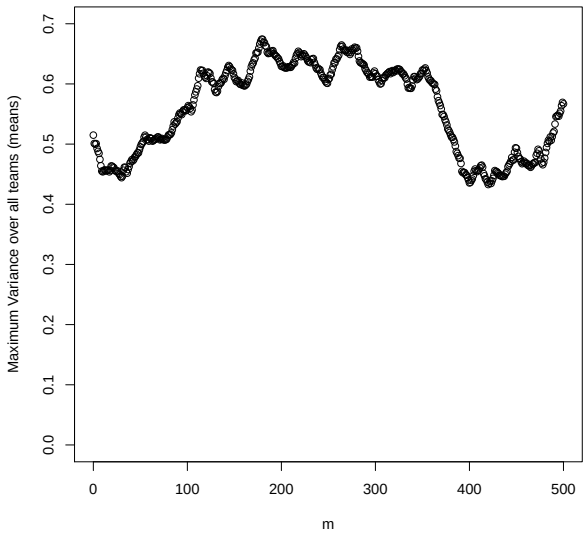
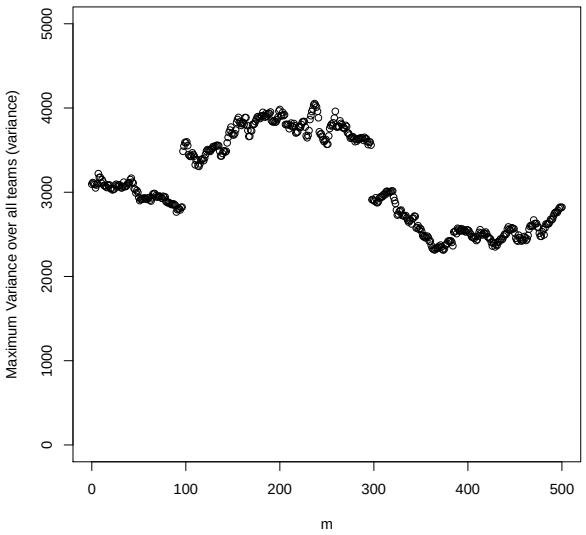


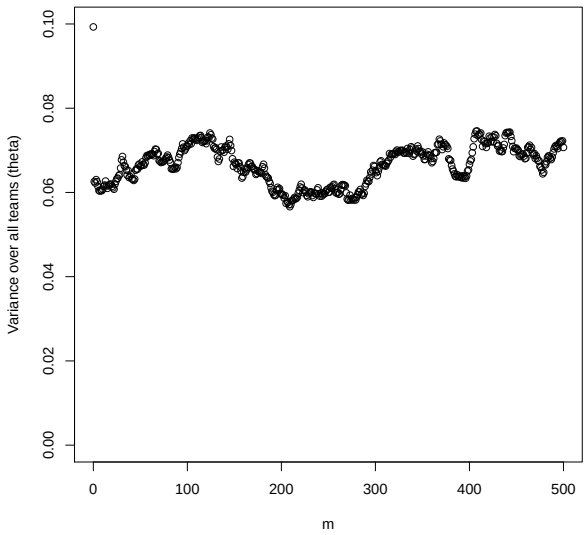
Figure 5: Graph showing the performance of Gibbs sampler to estimate the posterior distribution of θ for 5 different starting values, compared to the distribution shown in Figure 1



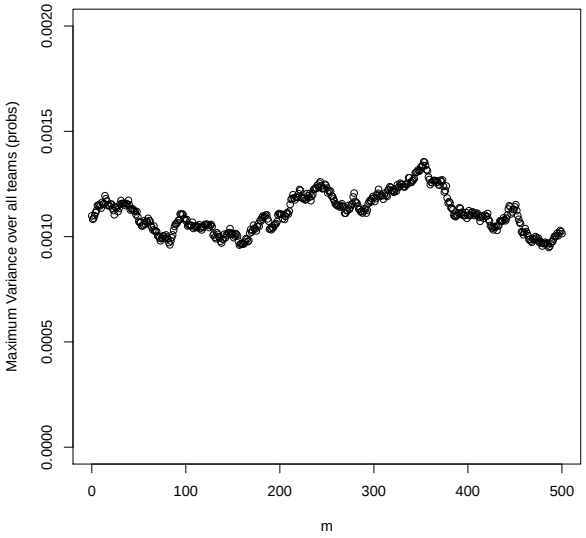
(a) Graph to show the maximum sample variance of Gibbs sampler approximation of the posterior mean of μ_k over all teams using m pre-iterations and 100 sample iterations



(b) Graph to show the maximum sample variance of Gibbs sampler approximation of the posterior mean of σ_k^2 over all teams using m pre-iterations and 100 sample iterations



(c) Graph to show the sample variance of Gibbs sampler approximation of the posterior mean of θ using m pre-iterations and 100 sample iterations



(d) Graph to show the maximum sample variance of Gibbs sampler approximation of the probability that a team is greater than average using m pre-iterations and 100 sample iterations

Figure 6: Graphs showing the accuracy of Gibbs Sampler in the approximations it makes, and how it varies depending on the number of pre-iteration performed

Program Listings

Code to set up the Gibbs Sampler, as well as the Gibbs Sampler

```
params <- rep(0, 5)
names(params) <- c("theta_0", "alpha_0", "beta_0", "mu_0", "tau_0")
params["sigma_0"] = 10
params["alpha_0"] = 1e-5
params["beta_0"] = 1e-3
params["mu_0"] = 60
params["tau_0"] = 20

gibbs_sampler <- function(data, start, iter, d_params) {
  # note that in the data the columns are different times
  # while the rows are different teams
  num_teams = dim(data)[1]
  num_years = dim(data)[2]

  sum_data_over_time = rowSums(data)

  result = matrix(c(start, rep(0,iter*(2*num_teams+1))),
                  ncol=2*num_teams+1, byrow = TRUE)
  for (i in 1:iter) {
    cur_sol = result[i, ]
    # Calculate all mu's
    for (k1 in 1:num_teams) {
      sigma_k_squared = cur_sol[num_teams+k1]
      dist_var = 1/(num_years*sigma_k_squared^(-1)+d_params["sigma_0"
         ]^(-2))
      dist_mu = (sum_data_over_time[k1]*sigma_k_squared^(-1) +
                   d_params["sigma_0"]^(-2)*cur_sol[2*num_teams+1])*
                      dist_var
      cur_sol[k1] = rnorm(1, mean = dist_mu, sd = sqrt(dist_var))
    }
    # Calculate all sigma
    faux_var = rowSums((data-cur_sol[1:num_teams])^2)
    for (k2 in 1:num_teams) {
      alpha_i = d_params["alpha_0"] + num_years/2
      beta_i = d_params["beta_0"] + (1/2)*faux_var[k2]
      cur_sol[num_teams + k2] = 1/rgamma(1, shape = alpha_i, rate =
         beta_i)
    }
    # calculate theta
    dist_var = 1/(num_teams*d_params["sigma_0"]^(-2)+d_params["tau_0"
       ]^(-2))
    dist_mu = (d_params["sigma_0"]^(-2)*sum(cur_sol[1:num_teams])-
                 d_params["mu_0"]*d_params["tau_0"]^(-2))*dist_var
    cur_sol[2*num_teams + 1] = rnorm(1, mean = dist_mu, sd = sqrt(dist
       _var))
    result[i+1, ] = cur_sol
  }
  # Just for my own readability
  result_colnames = c(
    unname(sapply(rownames(data), function(x){paste(x, "mean")})),
```

```
    unname(sapply(rownames(data), function(x){paste(x, "variances")}))
      ,
    "theta"
  )
  colnames(result) <- result_colnames
  return(result)
}


# We made the starting point estimations of each of the parameters
# i.e starting means is the mean score of team
starting_means = rowMeans(Football)
# starting vars is the vars of each team according to the data
starting_vars = rowMeans(Football^2)-starting_means^2
# starting theta is the mean of all scores
starting_theta = mean(starting_means) # valid as all rows have same
   number of entries
gibbs_start = c(starting_means, starting_vars, starting_theta)
gibbs_start = unname(gibbs_start)

iterations = 10000

result = gibbs_sampler(Football, gibbs_start, iterations, params)
K = dim(Football)[1]
```

Code to generate all outputs in Question 4

```
post_mean = colMeans(result)
mean_strings = paste("The posterior mean of",names(post_mean), ":",
   post_mean)
write.table(mean_strings, file = "means.txt", sep="\n", row.names =
   FALSE, col.names = FALSE, quote=FALSE)
post_theta_samples = result[, 2*K+1]
pdf("post_theta_dist.pdf")
post_theta_hist_info = hist(post_theta_samples, freq = F, xlab=TeX("$
   \\theta$"), main = TeX("Histogram of the posterior distribution of
   $\\theta$"), ylim=c(0,0.14))
dev.off()
```

Code to generate all outputs in Question 5

```
apply(result, 1, function(x) {as.numeric(x[1]>x[2*K+1])})
prob_mu_gt_theta = colMeans(t(apply(result, 1, function(x) {x[1:K]>x[2
   *K+1]})))
names(prob_mu_gt_theta) <- rownames(Football)
prob_strings = paste("Probability mean score of", names(prob_mu_gt_
   theta),
                     "is greater than average is:", prob_mu_gt_theta)
write.table(prob_strings, file = "probs.txt", sep="\n", row.names =
   FALSE, col.names = FALSE, quote=FALSE)
```

Code to generate all outputs in Question 6

```
N = 3000
trials = 100
trial_results = array(0, dim = c(N+1, 2*K+1, trials))
for (i in 1:trials) {
  trial_results[,,i] = gibbs_sampler(Football, gibbs_start, N, params)
}

# Points represents the number of points I want to sample e.g. If I
   want to sample at
# 5 different values of N from N = 2 to 1001, then points = 5
points = 40
var_mat_probs = matrix(0, nrow = points, ncol = K)
var_mat_mus = matrix(0, nrow = points, ncol = K)
colnames(var_mat_probs) <- rownames(Football)
# n_seq is a list representing the stopping point of several trials
   that is to simulate 5
# trials with N = 5, 10, 15, 20, can do 1 trial of length 20, then
   just look at the data points
# up until N = 5, 10, 15, 20
n_seq = floor(seq(2,N+1, length.out = points))

for (i in 1:points) {
  n = n_seq[i]
  # Here we are approximating P(mu_k > theta) with the given data (i.e
     iterations 1 to n)
  probs_mu_gt_theta = apply(trial_results[1:n,,], 3, function(x) {
  colMeans(t(apply(x, 1, function(z) {
    z[1:K]>z[2*K+1]
    })))
  })
  # Here we are approximating mu_k with the given data (i.e iterations
     1 to n)
  mus = apply(trial_results[1:n,,], 3, function(x) {
    colMeans(x[, 1:K])
  })

  var_probs_mu_gt_theta = rowMeans(probs_mu_gt_theta^2)-rowMeans(probs
     _mu_gt_theta)^2
  var_mat_probs[i,] = var_probs_mu_gt_theta
  var_mus = rowMeans(mus^2)-rowMeans(mus)^2
  var_mat_mus[i,] = var_mus
}

# We could look at indvidual team k, but we could also look at all
   teams then consider the
# maximum variance, thus looking at some kind of uniform convergence
pdf("var_probs.pdf")
max_var_probs = apply(var_mat_probs, 1, max)
plot(n_seq, max_var_probs, type="l",
     xlab="n", ylab="Maximum Variance over all teams (probabilities)")
dev.off()

pdf("var_means.pdf")
```

```
max_var_mus = apply(var_mat_mus, 1, max)
plot(n_seq, max_var_mus, type="l",
     xlab="n", ylab="Maximum Variance over all teams (means)")
dev.off()


# Now to investigate the relation between the variance and n
pdf("log_var_probs.pdf")
log_max_var_probs = log(max_var_probs)
plot(n_seq, log_max_var_probs, type="l",
     xlab="n", ylab="Log Maximum Variance over all teams (
        probabilities)")
dev.off()

pdf("log_var_means.pdf")
log_max_var_mus = log(max_var_mus)
plot(n_seq, log_max_var_mus, type="l",
     xlab="n", ylab="Log Maximum Variance over all teams (means)")
dev.off()
```

Code to generate all outputs in Question 7

```
min_M = 100
max_M = 3000
num_M = 5
M_seq = floor(seq(min_M, max_M, length.out = num_M))
N = 1000
trials = 100
trial_results = array(0, dim = c(N+max_M+1, 2*K+1, trials))
for (i in 1:trials) {
  trial_results[,,i] = gibbs_sampler(Football, gibbs_start, N+max_M,
     params)
}

points = 40
var_mat_probs = array(0, dim = c(points, num_M, K))
var_mat_mus = array(0, dim = c(points, num_M, K))
n_seq = floor(seq(2,N+1, length.out = points))
for (i in 1:points) {
  for (j in 1:num_M) {
    n = n_seq[i]
    m = M_seq[j]
    probs_mu_gt_theta = apply(trial_results[(m+1):(m+n),,], 3,
       function(x) {
      colMeans(t(apply(x, 1, function(z) {
        z[1:K]>z[2*K+1]
      })))
    })
    mus = apply(trial_results[(m+1):(m+n),,], 3, function(x) {
      colMeans(x[, 1:K])
    })
    var_probs_mu_gt_theta = rowMeans(probs_mu_gt_theta^2)-rowMeans(
       probs_mu_gt_theta)^2
    var_mat_probs[i, j, ] = var_probs_mu_gt_theta
```

```
    var_mus = rowMeans(mus^2)-rowMeans(mus)^2
    var_mat_mus[i, j, ] = var_mus
  }
}

# We could look at indvidual team k, but we could also look at all
   teams then consider the
# maximum variance, thus looking at some kind of uniform convergence
cols = rainbow(num_M)
pdf("var_probs_2.pdf")
matplot(t(matrix(rep(n_seq,num_M), nrow=num_M, byrow = T)),
        apply(var_mat_probs, 2, function(x) {apply(x, 1, max)}), type=
           "l",
        col = cols, xlab="n", ylab="Maximum Variance over all teams (
           probabilities)")
legend("topright", legend = M_seq, col=cols, lwd=2, ncol=2, cex=0.8,
   title = "Number, M,  of pre-iterations")
dev.off()

pdf("var_means_2.pdf")
matplot(t(matrix(rep(n_seq,num_M), nrow=num_M, byrow = T)),
        apply(var_mat_mus, 2, function(x) {apply(x, 1, max)}), type="l
           ",
     xlab="n", ylab="Maximum Variance over all teams (means)")
legend("topright", legend = M_seq, col=cols, lwd=2, ncol=2, cex=0.8,
   title = "Number, M,  of pre-iterations")
dev.off()


min_M = 1
max_M = 1000
num_M = 100
M_seq = floor(seq(min_M, max_M, length.out = num_M))
N = 100
trials = 100
trial_results = array(0, dim = c(N+max_M+1, 2*K+1, trials))
for (i in 1:trials) {
  trial_results[,,i] = gibbs_sampler(Football, gibbs_start, N+max_M,
     params)
}

points = 4
var_mat_probs = array(0, dim = c(points, num_M, K))
var_mat_mus = array(0, dim = c(points, num_M, K))
# colnames(var_mat_probs) <- rownames(Football)
n_seq = round(exp(seq(log(2),log(N), length.out=points)))
for (i in 1:points) {
  for (j in 1:num_M) {
    n = n_seq[i]
    m = M_seq[j]
    probs_mu_gt_theta = apply(trial_results[(m+1):(m+n),,], 3,
       function(x) {
      colMeans(t(apply(x, 1, function(z) {
```

```
        z[1:K]>z[2*K+1]
      }))))
    })
    mus = apply(trial_results[(m+1):(m+n),,], 3, function(x) {
      colMeans(x[, 1:K])
    })
    var_probs_mu_gt_theta = rowMeans(probs_mu_gt_theta^2)-rowMeans(
       probs_mu_gt_theta)^2
    var_mat_probs[i, j, ] = var_probs_mu_gt_theta
    var_mus = rowMeans(mus^2)-rowMeans(mus)^2
    var_mat_mus[i, j, ] = var_mus
  }
}

cols = rainbow(points)
pdf("var_probs_3.pdf")
matplot(M_seq,
        apply(var_mat_probs, 1, function(x) {apply(x, 1, max)}), type=
           rep("p", points),
        col = cols, xlab="m", ylab="Maximum Variance over all teams (
           probabilities)", pch=rep(1, points))
legend("right", bg="white", legend = n_seq, col=cols, pch=rep(1,
   points), ncol=2, cex=0.8, title = "Number, n, of iterations")
dev.off()

pdf("var_means_3.pdf")
matplot(M_seq,
        apply(var_mat_mus, 1, function(x) {apply(x, 1, max)}), type=
           rep("p", points),
        col = cols, xlab="m", ylab="Maximum Variance over all teams (
           means)", pch=rep(1, points))
legend("topright", bg="white", legend = n_seq, col=cols, pch=rep(1,
   points), ncol=2, cex=0.8, title = "Number, n, of iterations")
dev.off()
```

Code to generate all outputs in Question 6

```
print("Question 8")
num_start_points = 200
iterations=200
max_pre_iterations = 500
num_theta_analysis = 5

starting_points = matrix(
  c(runif(K*num_start_points, 0, 120),
    runif(K*num_start_points, 0.001,40),
    runif(num_start_points, 0, 120)), nrow = 2*K+1, byrow = TRUE)

all_data = array(0, dim=c(num_start_points, max_pre_iterations+
   iterations+1, 2*K+1))

for (i in 1:num_start_points) {
  data = gibbs_sampler(Football, starting_points[, i], max_pre_
     iterations + iterations, params)
```

```
  all_data[i, , ]=data
}

write.table(all_data[1:5,1:5 ,2*K+1 ], file = "post_theta_iters_1.txt"
   , row.names=FALSE, col.names=FALSE)
write.table(all_data[1:5,(iterations - 3):(1+iterations) ,2*K+1 ],
   file = "post_theta_iters_2.txt", row.names=FALSE, col.names=FALSE)

pdf("theta_dist_conv.pdf")
plot(post_theta_hist_info, freq=FALSE, ylim = c(0, 0.14),
     xlab=TeX("$\\theta$"), main = TeX("Histogram of the posterior
        distribution of $\\theta$"))
cols = rainbow(num_theta_analysis)
for (i in 1:num_theta_analysis) {
  break_points = seq(floor(min(all_data[i,1:(1+iterations) , 2*K+1]))
     - (floor(min(all_data[i,1:(1+iterations) , 2*K+1]))%%2),
                     ceiling(max(all_data[i,1:(1+iterations) , 2*K+1])
                        ) + (ceiling(max(all_data[i,1:(1+iterations) ,
                         2*K+1]))%%2),
                     by=2)
  hist_data = hist(all_data[i,1:(1+iterations), 2*K+1], plot = FALSE,
     breaks = break_points)
  lines(hist_data$mids, hist_data$density, col=cols[i])
}
dev.off()

means = t(apply(all_data[,1:(1+iterations),], 1, function(x) {colMeans
   (x)}))
probs = t(apply(all_data[,1:(1+iterations),], 1, function(x) {
  rowMeans(
    apply(x, 1, function(z) {
      z[1:K] > z[2*K + 1]
    })
  )
}))

all_estimates = cbind(means, probs)
all_variance = colMeans(all_estimates^2)-colMeans(all_estimates)^2
max_sds = c(max(all_variance[1:K]),
            max(all_variance[(K+1):(2*K)]),
            max(all_variance[2*K+1]),
            max(all_variance[(2*K+2):(3*K+1)]))^0.5

write.table(c(paste("Maximum standard deviation for posterior mean of
   mu_k is", max_sds[1]),
              paste("Maximum standard deviation for posterior mean of
                 sigma_k^2 is", max_sds[2]),
              paste("Maximum standard deviation for posterior mean of
                 theta is", max_sds[3]),
              paste("Maximum standard deviation for posterior
                 probability mu_k > theta is", max_sds[4]))
          ,file = "max_estimate_sds.txt", sep="\n", row.names = FALSE,
            col.names = FALSE, quote=FALSE)
```

```
write.table(t(rbind(colMeans(all_estimates)[(K+1):(2*K)], all_variance
   [(K+1):(2*K)]^0.5)),
            file = "sigma_sds.txt", sep=" ", row.names = FALSE, col.
               names = c("Expected value", "Standard Deviation"),
               quote=FALSE)

pre_iter_max_vars = matrix(rep(c(0,0,0,0), max_pre_iterations+1), nrow
   =max_pre_iterations+1)

for (i in 0:max_pre_iterations) {
  means = t(apply(all_data[,(i+1):(i+1+iterations),], 1, function(x) {
     colMeans(x)}))
  probs = t(apply(all_data[,(i+1):(i+1+iterations),], 1, function(x) {
    rowMeans(
      apply(x, 1, function(z) {
        z[1:K] > z[2*K + 1]
      })
    )
  }))

  all_estimates = cbind(means, probs)
  all_variance = colMeans(all_estimates^2)-colMeans(all_estimates)^2
  max_vars = c(max(all_variance[1:K]),
               max(all_variance[(K+1):(2*K)]),
               max(all_variance[2*K+1]),
               max(all_variance[(2*K+2):(3*K+1)]))
  pre_iter_max_vars[i+1, ] = max_vars
}

pdf("max_var_estimates_1.pdf")
plot(0:max_pre_iterations, pre_iter_max_vars[, 1], ylim=c(0,0.7), pch
   =1, xlab="m", ylab="Maximum Variance over all teams (means)")
dev.off()
pdf("max_var_estimates_2.pdf")
plot(0:max_pre_iterations, pre_iter_max_vars[, 2], ylim=c(0,5000), pch
   =1, xlab="m", ylab="Maximum Variance over all teams (variance)")
dev.off()
pdf("max_var_estimates_3.pdf")
plot(0:max_pre_iterations, pre_iter_max_vars[, 3], ylim=c(0,0.1), pch
   =1, xlab="m", ylab="Variance over all teams (theta)")
dev.off()
pdf("max_var_estimates_4.pdf")
plot(0:max_pre_iterations, pre_iter_max_vars[, 4], ylim=c(0,0.002),
   pch=1, xlab="m", ylab="Maximum Variance over all teams (probs)")
dev.off()
```