

Q1

For the actual Implementation, please refer to the R code script.

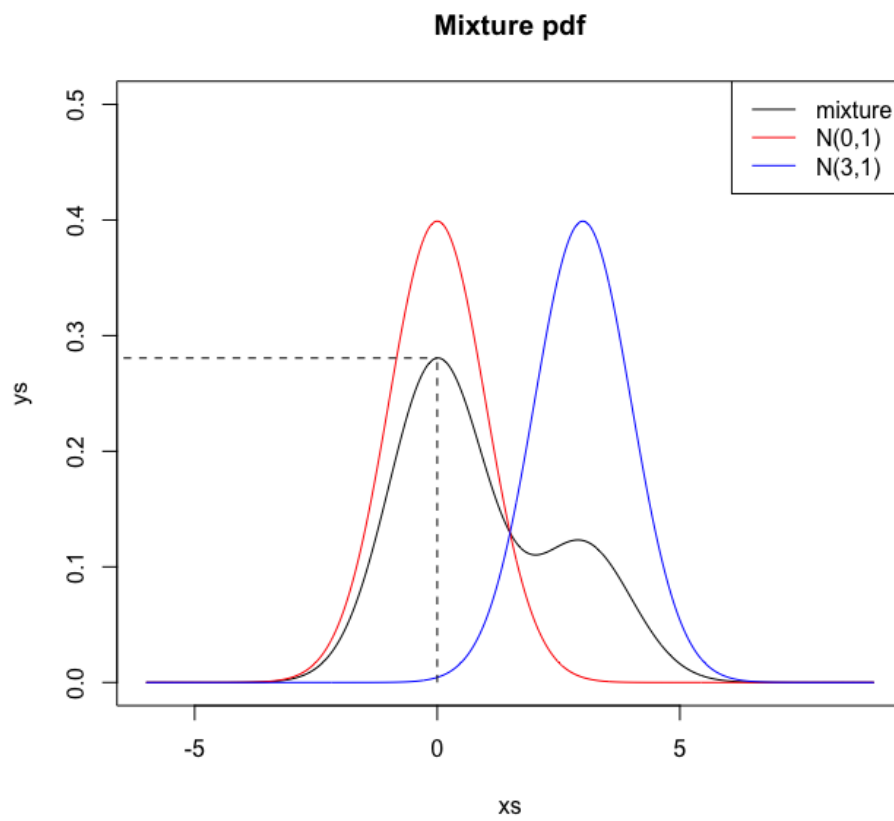
From my R-scripts, I performed the following things:

- i) define a function for the pdf of mixture distribution
- ii) get the true value of theta (i.e. the mixture pdf at $x = 0$)
- iii) define a function to simulate from the mixture distribution
- iv) define a function for KDE (and can perform the same functionality as **density(...)**, which allow me to evaluate the value at $x=0$)
- v) following the coursework requirements, to implement the nonparametric bootstrap method.

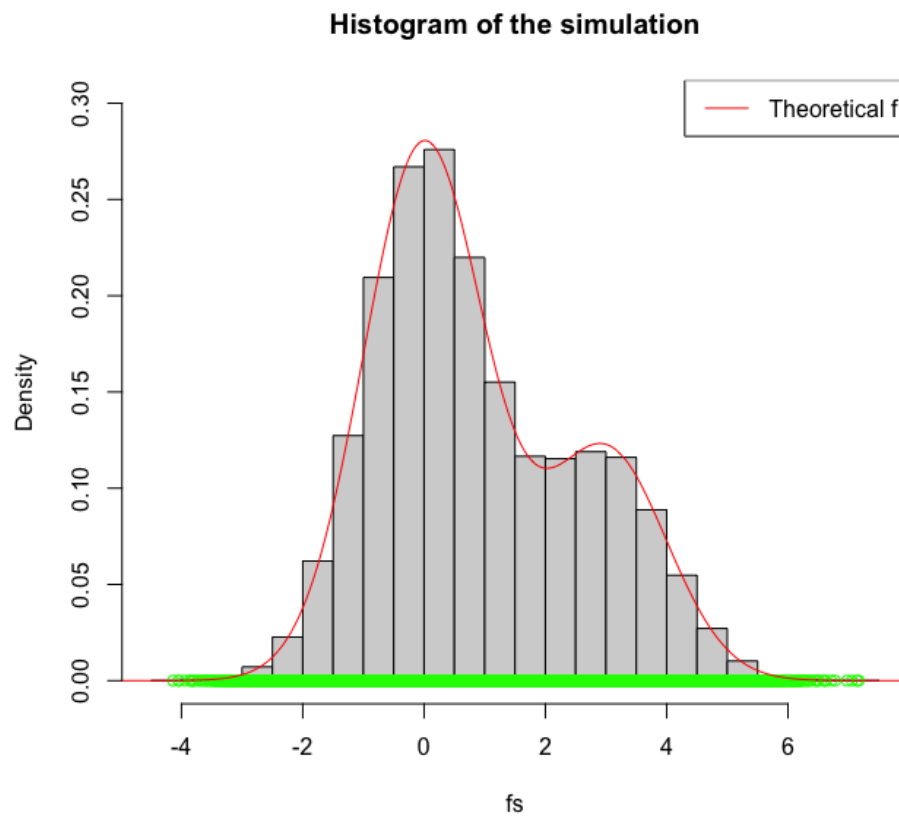
And the actual nonparametric bootstrap method outlines:

- i) prepare 100 different bandwidth (h) for KDE
- ii) for each h, run J=200 times in order to estimate the coverage
- iii) for each run (j from 1 to 200)
 - a) simulate $N = 100$ data points from the mixture
 - b) perform $B=1000$ bootstrap replicates, using the same set of simulated data points
 - 1) for each Bootstrap replicates (b from 1 to 1000)
- 2) using those simulated data points to run KDE with bandwidth=h, and
evaluated at $x=0$, called this estimated theta
- 3) store this estimated theta
- 4) after these 1000 bootstrap replicates, determine a 95% percentile confidence interval
- 5) check if this interval contains the true theta
- c) from these $B=1000$ different intervals, and we know how many intervals contains the true theta, calculate the ratio (estimated coverage)
- iv) plot the estimated coverage against h

The plot of the mixture pdf



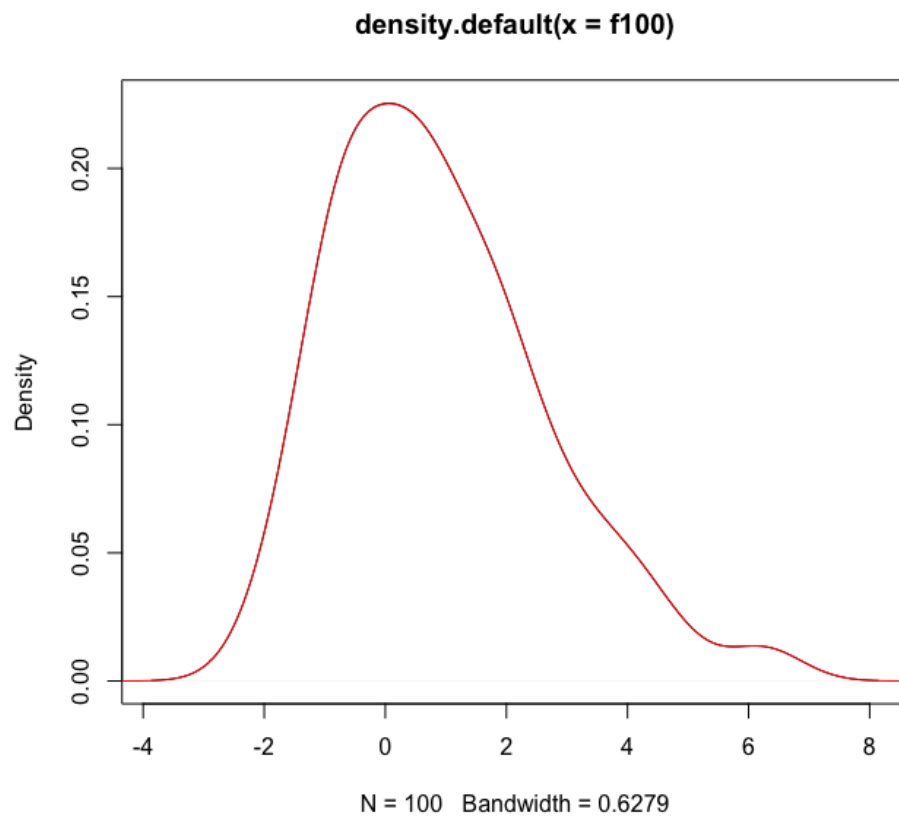
To check if my function to simulate from the mixture works



To check if my KDE function works.

I simulated 100 data points from the mixture, and then first use R built-in **density** command to get its suggested bandwidth, and then I use my KDE function to plot my KDE function using this bandwidth. And finally, overlay using built-in **density**

Which is perfect match as expected



From the given instructions for the simulation study, I will use

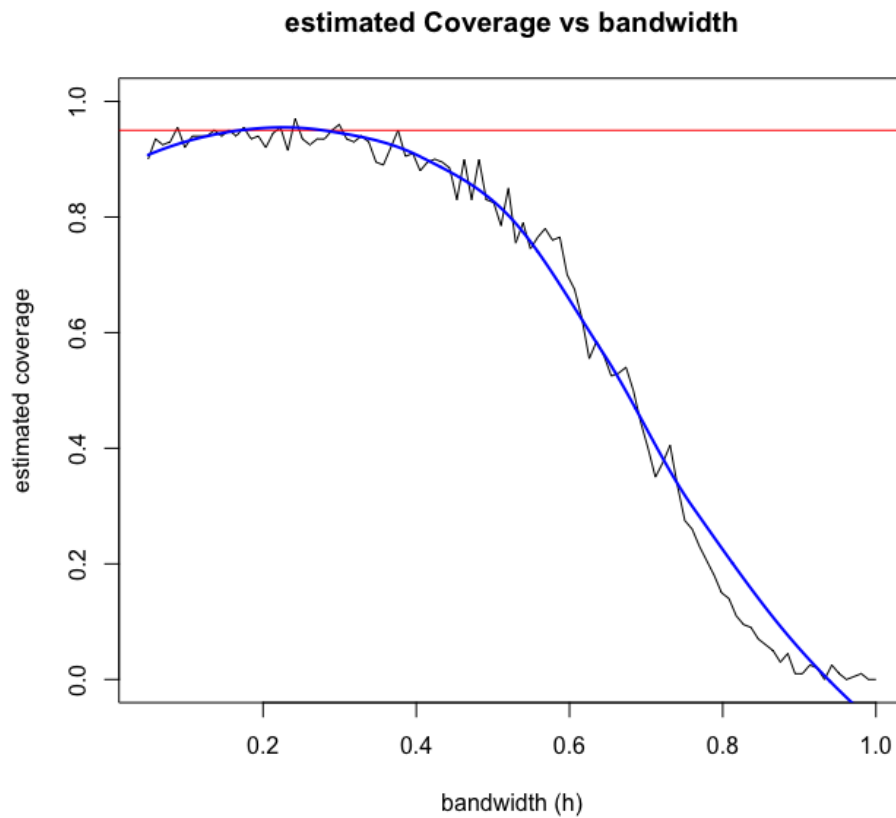
- 100 simulated data point from the mixture distribution,
- bootstrap sample size = 100

I choose to run 1000 times to get 95% percentile confidence interval, and run 200 times to find the estimated coverage.

First run using some fixed bandwidth.

I used $h = 0.5$, and the estimated coverage is 0.84

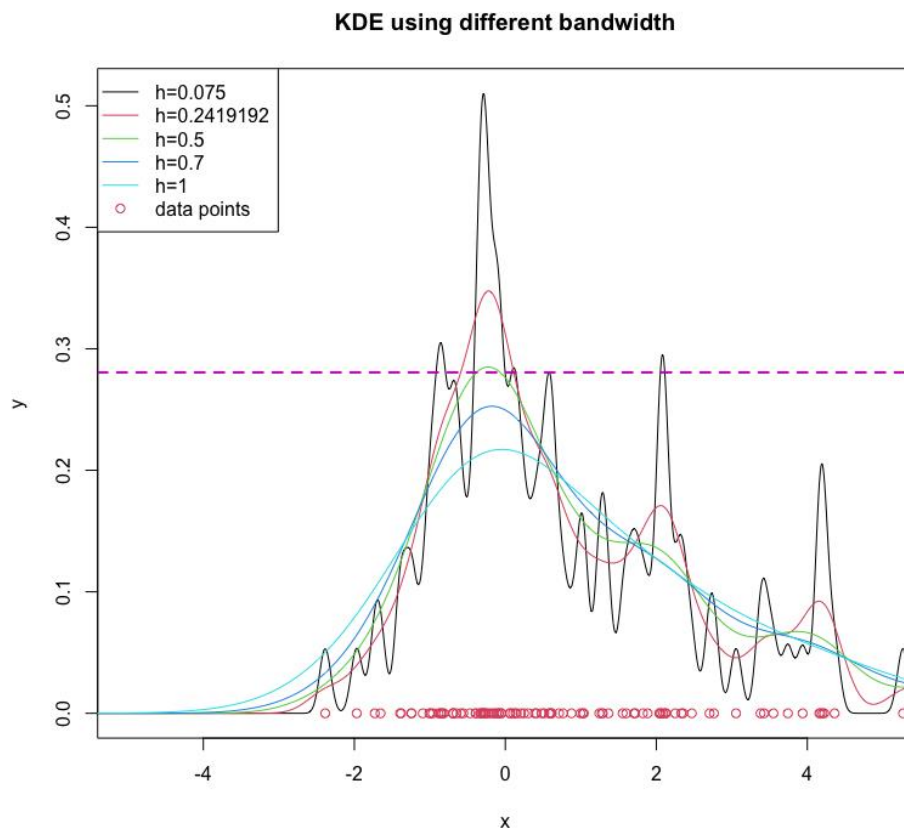
Next, I run for various bandwidth, the below plot is the estimated coverage varies with different bandwidth



From this plot, we can see the estimated coverage is almost above 0.9 when h is between 0.05 and 0.4, and decreases quickly when h is between 0.4 to 1. And it almost 0 when $h \geq 0.9$

We also see that there seems to be a maximum at $h = 0.2419$, where its estimated coverage is 0.97.

To get some insight about the above plots, we can make some sample KDE plots, using different bandwidth with some simulated data points



Recall that when using kernel density estimation, the density will concentrate on each data point if the bandwidth is small; and in contrast, if the bandwidth is large, then the density will be "flattened". This is helpful to explain the previous plot.

If h is very small, then the KDE highly concentrated on each data point. Because our mixture distribution has high density around 0, so from our simulation study, it is quite possible to simulate data around 0, and quite possible that the bootstrap sample includes some data point around 0. But anyway, like the above plot for $h = 0.075$, because it is possible that too many data point around 0 to make the KDE shows "extreme spike" at $x = 0$, so the estimate will then be far above the true theta (hence, the 95% percentile confidence interval will not cover the true theta for some cases). In general for h is very small, it still has about 90% on average to coverage to true theta.

If h is very large, then the KDE is almost flattened, and by this flattened nature, the KDE y-value is almost impossible to close to the true theta at $x = 0$ (almost always below the true theta).

Therefore, in between those, for around $h = 0.24$, that shows some balance between the two extremes, neither highly fluctuating like h is very small; nor very flattened like $h = 1$. So, this such h gives the most capability to include the true θ at $x = 0$, and so the estimated coverage is the highest.

Q2

2a) Refer to handwritten work

2b) Refer to handwritten work

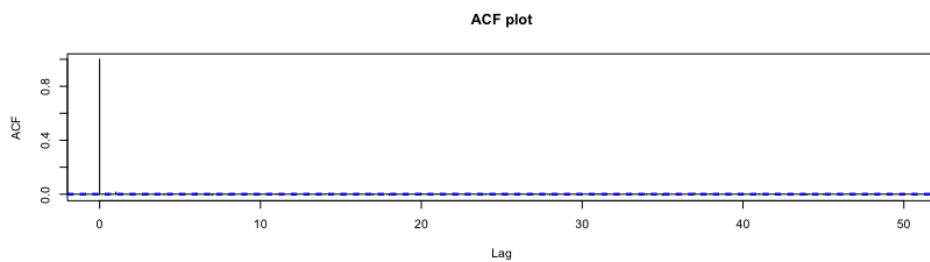
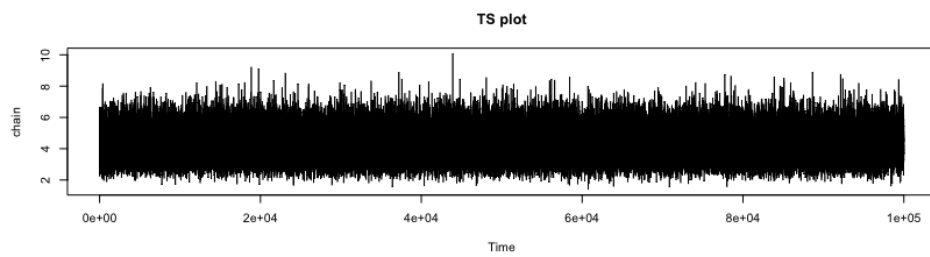
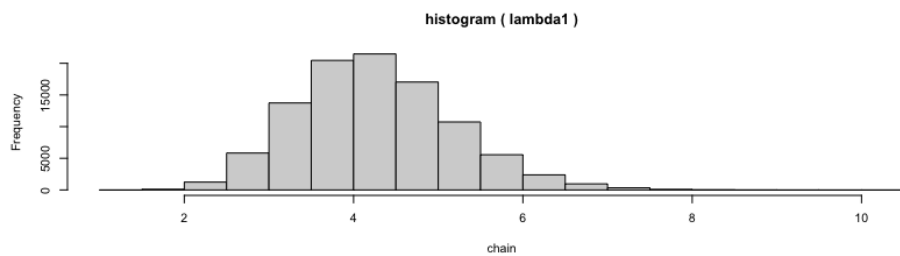
2c) Refer to R code script

2d)

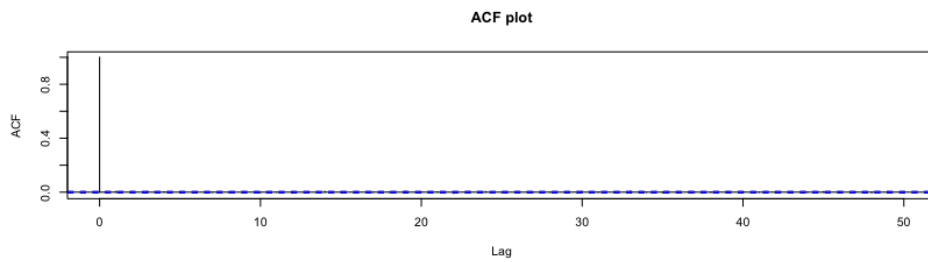
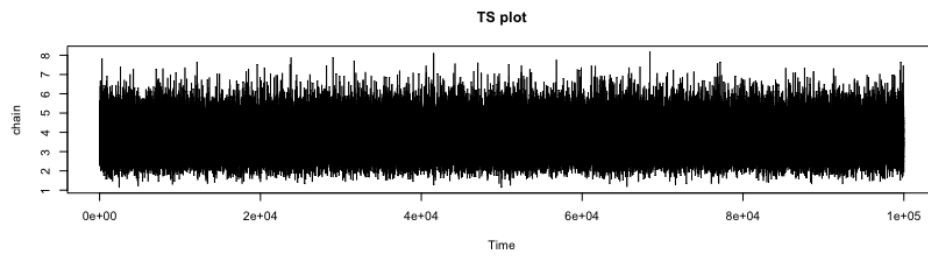
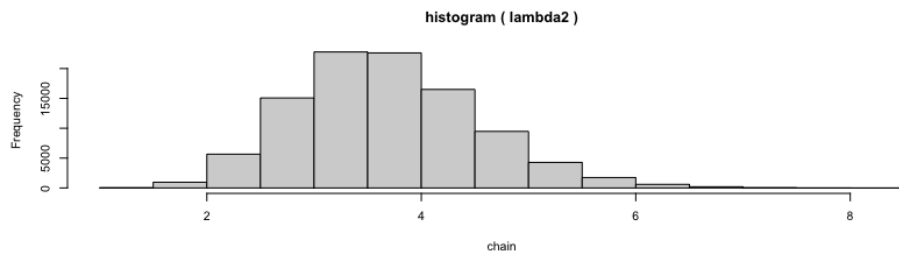
I ran $N = 100000$ iterations for the Gibbs-sampler algorithm.

The followings are the plots for each variable:

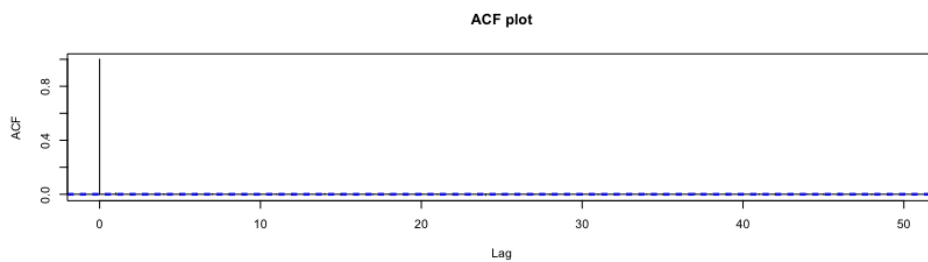
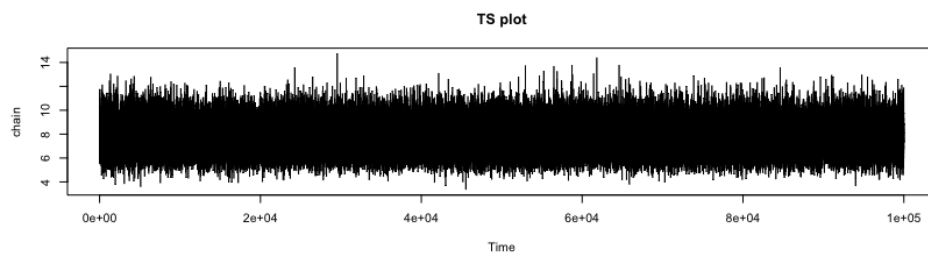
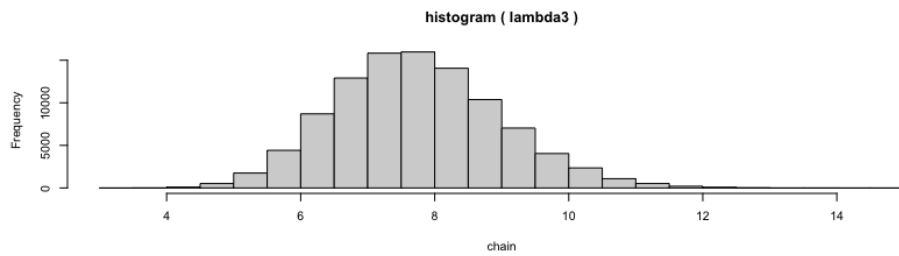
λ_1 :



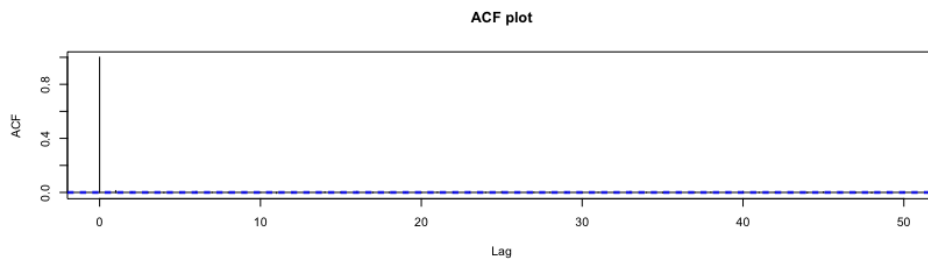
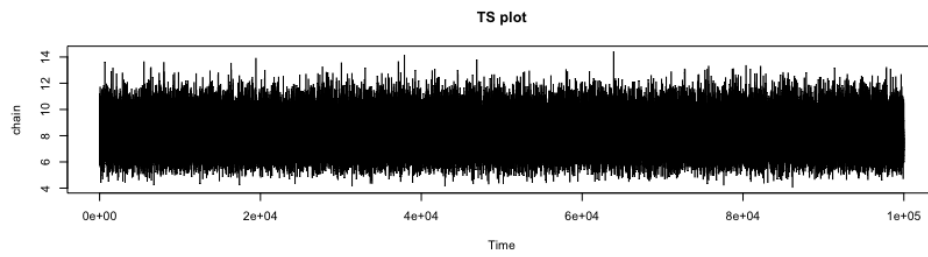
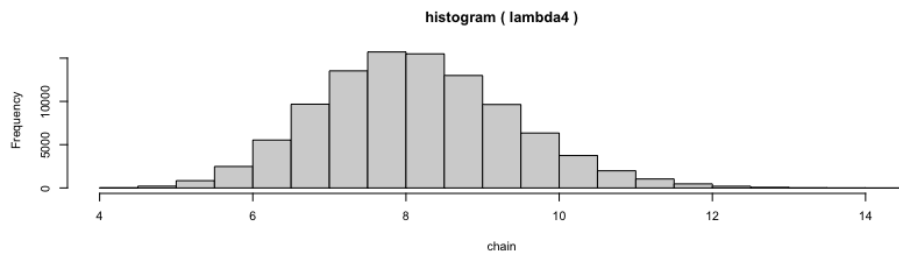
λ_2 :



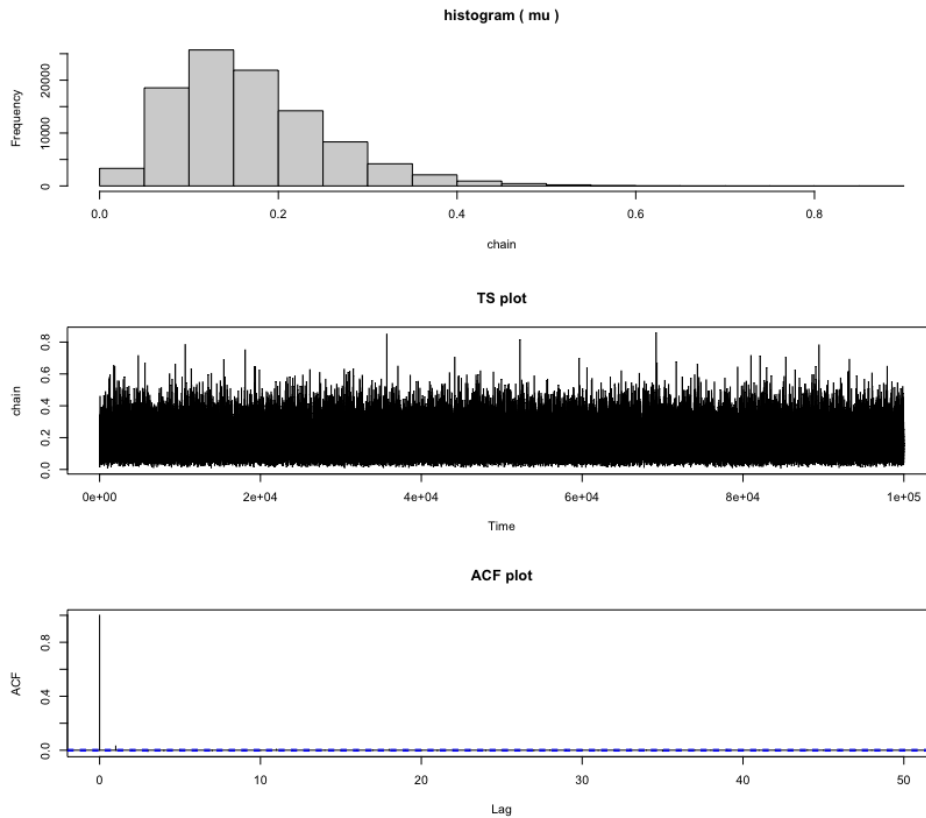
λ_3 :



λ_4 :



μ :



Since for all 5 variables above, the trace plot (I.e. the time-series plot of the chain) looks reached stationary distribution; and the ACF plot reaches to 0 very quickly.

They are evidence that the MCMC mix well.

p.s. I tried different initial starting point of μ give similar output.

2e)

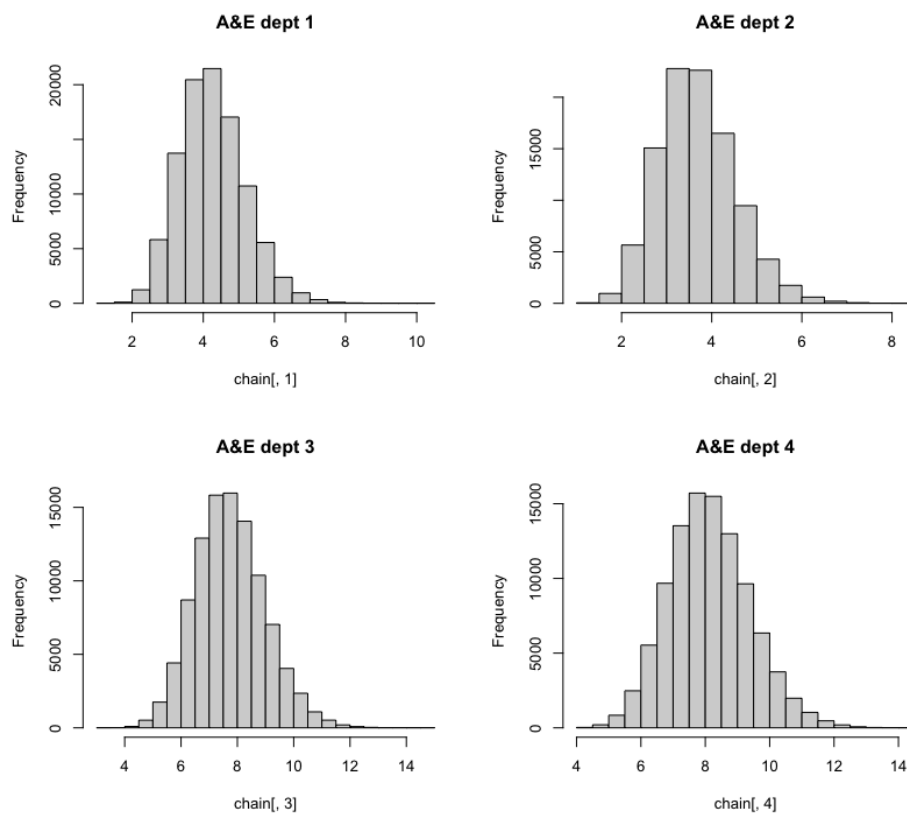
Calculate the posterior mean for each 4 λ s

	λ_1	λ_2	λ_3	λ_4
Posterior mean	4.258668	3.680244	7.743447	8.126413
Posterior variance	0.8358464	0.7187025	1.51826	1.587527

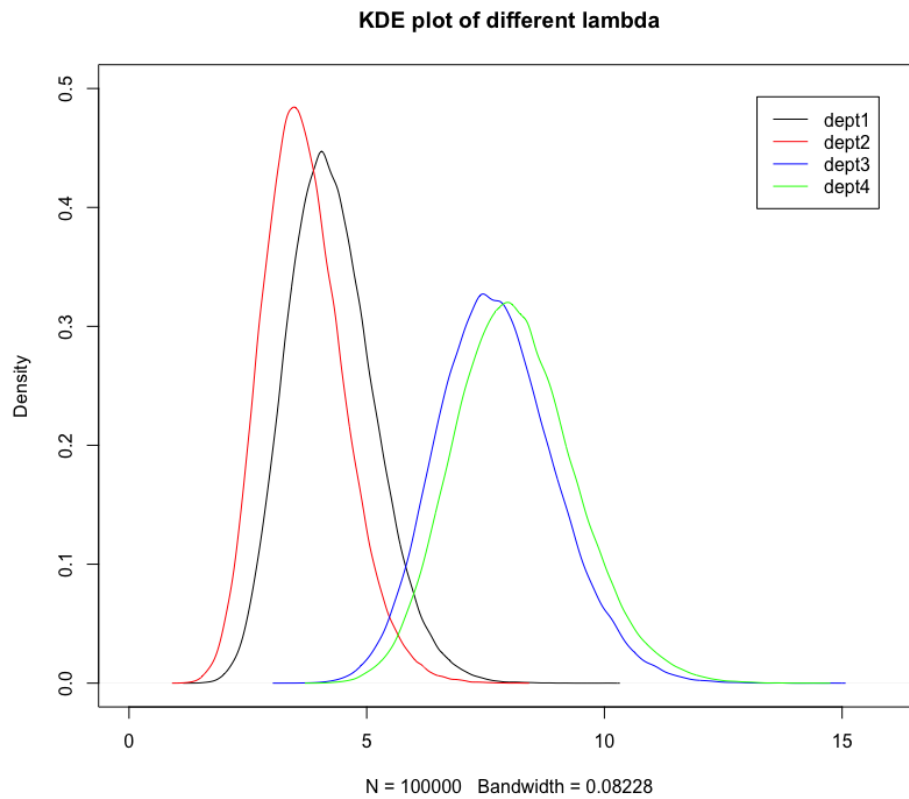
We can see that the posterior mean of λ_4 is the largest, so A&E department 4 has the highest night admission rates.

Also, the posterior variance of λ_4 is also the largest, so A&E department 4 has the highest variability in night admission rates.

In contrast, A&E department 2 has the smallest posterior mean and variance of λ_2 .



Because I didn't derive the marginal posterior distribution of each λ_s , but we can still use KDE to get a smoothed curve, and plot together



From the plot above, we can see that, the density of dept1 and dept2 looks similar; and dept3 and 4 looks similar.

p.s. note that the value of each posterior mean of λ 's are quite make-sense, because,

For each i from the given relationship,

$$y_{ij} \mid \lambda_i, \mu \sim \text{Po}(\lambda_i)$$

$$\text{So, } E(y_{ij} \mid \lambda_i, \mu) = E(\text{Po}(\lambda_i)) = \lambda_i$$

From our raw data, the mean of each dept is

dept1	dept2	dept3	dept4
4.2	3.6	7.8	8.2

Where our posterior means are quite close to the mean of original observations for each dept, as I expected.

2f)

From the output of the simulation, we can easily obtain the sample of marginal posterior distribution of the variable λ_4 by extracting the column for λ_4 alone.

i.e. we can obtain sample from $\pi(\lambda_4 \mid \mathbf{y})$ from the output of the MCMC.

In addition, we know that $y \mid \lambda_4 \sim \text{Po}(\lambda_4)$, so we can sample new y from $\text{Po}(\lambda_4)$

Now, suppose we have N samples from $\pi(\lambda_4 \mid \mathbf{y})$, let λ_{4_i} be the i -th sample,

then we can simulate y by using the fact that y^* and y have the same distribution, so, $y^* \mid \lambda_4 \sim \text{Po}(\lambda_4)$.
So, the simulated y , say y_i = the simulated value from λ_{4_i}

hence, we can obtain a collection of pairs

$(\lambda_{4_1}, y^*_{_1})$

$(\lambda_{4_2}, y^*_{_2})$

....

$(\lambda_{4_N}, y^*_{_N})$

these are the sample from the $\pi(\lambda_4, y^* \mid \mathbf{y})$

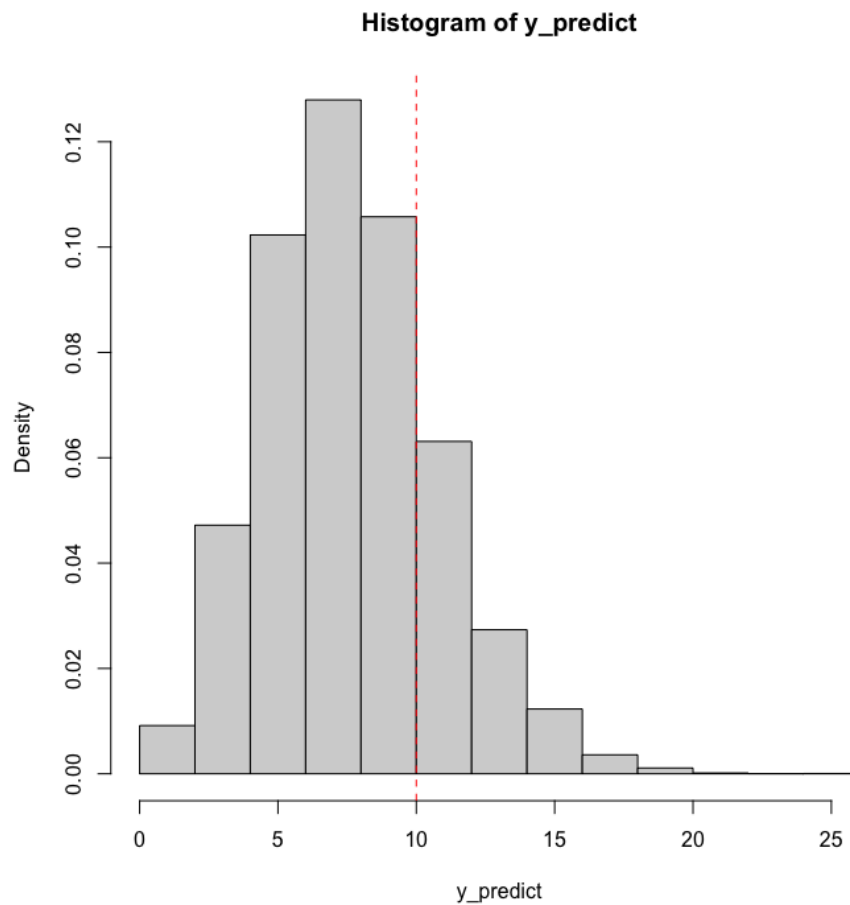
Then by just selecting the $\{y^*_{_i}\}$, form the sample of the predictive distribution of $y^* \mid y$

2g)

Refer to the R code to see the implementation.

Although there are seemed not having burn-in period, because the first few samples of the chain may be still depends on the initial starting point of μ . So for safety, I ignore the first 1000 data, and I draw 10000 samples from the remaining MCMC output, and from these 10000 samples of λ_4 , use rpoi to get 10000 samples from $\pi(y^* | \mathbf{y})$

Using these, I obtain the histogram of $\pi(y^* | \mathbf{y})$



And the required estimate of

$P(\text{the number of admissions to A\&E department 4 on this new night is at least } 10)$

$= 0.308$

