Jack Charbonneau
Jannik Haas
Jian Liu
Mario Arduz
Joan Wong

Project 2 Write Up - Group 1

Part 1. Kalman Filters

Goals:

1. Estimate the number of homeless people in Worcester
2. Construct transition and sensor models for this problem
3. From March 2020 through August 2020, create monthly reports with an estimate of the homeless population

OVERVIEW

We want to know the number of homeless people in Worcester every month.
What we have to work with:

1. A transition model of how the homeless population changes over time (monthly)
2. Observations of the number of people sheltered and the number of arrests for vagrancy in Worcester every month

Transition Model

Takes as input the homeless population at time, ***t***, and a predicted change in homelessness, ***a***

Actions = {
Decrease the number of homeless,
Increase the number of homeless,
No change,
}

Let y_t be the old (or *predicted*) homeless population size and $a \ \exists$ Actions

$y_{t+1} = y_t \times (1 + a)$

The monthly actions (in the year 2020) are presented below:

Month	**Action**
March	-16% *a = -0.16*
April	-9% *a = -0.09*
May	No change* *Note: For the purpose of our experiment, we would like to try out all possibilities of trend changes in the population size of homeless.* *a* = 0
June	+10% *a = 0.10*
July	+20% *a = 0.20*
August	+5% *a = 0.05*

According to the article [2], the number of homeless individuals has been growing by an average of 10% per year. Therefore, we expect the overall net change in population size of homeless to be +10% by the end of our experiment. We expect a decrease in the population size during the Spring time because we assume individuals who were sheltered in the past winter may have received enough assistance to change their homeless status (29% of the population were individuals and 86% of them were sheltered, which gives us about a net change of -25%) [3]. Additionally, based on the article [2], the number of individuals in shelters doubled in the past decade (assuming a constant change of +5% per year), and this trend will continue as shown in the report [3]. Therefore, near the end of our experiment (Summer to Fall), we expect the population size to have a net change of +34% (29% in the population were individuals + 5% per year). We will make this +35% for the ease of calculation.

The monthly, transitional actions (the change in population size of homeless displayed in the table above) have a variance of 140.88.

Thus, transition variance (sig_x) is around 140.

Sensor Model

We have 2 sensors for estimating the number of homeless at time ***t***
Assume in March 2020, the homeless population in Worcester is: $\mathbf{P_{t=March} = 1200}$

Since the estimate of the number of homeless people isn't perfect since there might be some homeless people that are well hidden or just became homeless and haven't been counted we have decided to use a standard deviation of 50 for our population size. Then we start with:

Population size (In February) = u = 1200
Population variance = sig_t = 50^2 = 2500

Sensor 1: Homeless convention

It is obvious that the number of people in homeless shelters is not equal to the actual number of homeless people in Worcester. We have decided to use the number of people in shelters as one of our sensors to estimate the number of homeless people in Worcester.

Let s = number of people in homeless shelters, y_S = population size of homeless in Worcester

According to the article [1], there are at least 1,100 homeless in Worcester. Hotel Grace has a maximum capacity of 50 and the Queen Street shelter has a capacity of 94 people for a total capacity of 144. We will use the number of people sheltered in these two shelters for our estimate of the homeless population. Assuming on average the shelters are 75% filled, that would result in 108 homeless people in shelters. With our current estimate of 1,100 homeless people in Worcester this results in a multiplier of about 11. Therefore,

$y_S = 11s$

Due to the fact that the number of homeless people in these shelters is highly variable on season and many other factors we estimate the standard deviation to be fairly high at 160.

[A] $y_S = 11s$ + (error with SD 160)

We made up a dataset for s and v based on the general thinking that more people will be in shelters and fewer arrests during colder times (e.g. March) as opposed to warmer times, where we would expect more people to be shelterless and thus more arrests.

Sensor 2: Arrest for vagrancy

Let v = number of arrests for vagrancy, y_V = population size of homeless in Worcester

We were not able to find relevant statistics on arrests for vagrancy in Worcester. Therefore, we will use Professor Beck's example on arrests for vagrancy making up 2% of the homeless population for the purpose of this assignment.

Assuming that we know 2% of homeless were arrested in any given month for vagrancy,

If v arrests for vagrancy is reported,

$y_V = (v / 0.02)$

The arrest rate of Massachusetts was reported at 1728.01 per 100,000 population [4] which backs up our claim of having about 2% of homeless people arrested for vagrancy in any given month. Let's assume that the number of arrests for vagrancy for a given month is 20. Using the equation above, we estimate the homeless population size of homeless for that month to be 1000. We expect this sensor to be highly variable since it depends on how many homeless people are sheltered and a large number of other factors, therefore we have decided to go with a standard deviation of 100. Therefore,

[B] $y_V = (v / 0.02) + (\text{error with SD } 100)$

We made up a dataset for s (number of people in shelters) and v (number of vagrancy arrests) based on the general thinking that more people will be in shelters and therefore we will see fewer arrests during colder times (e.g. March) as opposed to warmer times, where we would expect more people to be shelterless since they are able to sleep outside and thus more arrests.

Procedure

Please refer to the Excel sheet for more detail.

1. Estimate, y_S, from Sensor 1
2. Estimate, y_V, from Sensor 2
3. Combine the sensors to have a better estimate BE

   Because both sensor measurements are independent,

- Sensor reading, *combined_sensor_reading* = $c(y_S) + (1 - c)y_V$, where c and $(1 - c)$ are weights for sensors 1 and 2, respectively. We decided on a value of 0.4 for c since we believe the vagrancy arrests to be a slightly better indicator for the number of homeless people

- Overall sensor variance = *sig_z* = var(y_S) + var(y_V)

4. Combine BE with transition model estimate for new position

   The Kalman Gain **B** can be calculated by:

   B = (sig_t + sig_x)/(sig_t + sig_x + *sig_z*)

   New Population = (1 - **B**) × y_{t+1}+ **B** × (*combined_sensor_reading*)
   New Variance = (sig_t + sig_x) × sig_z / (sig_t + sig_x + sig_z)

At the end of your internship in August, your boss wants a final report containing two items:

1. A revised estimate of the homeless population for March 2020 - August 2020. Since you have 6 months of data, you should be able to better estimate the population for the intervening months. Explain the process for how you computed your revised estimate for April 2020. **ETA: For which months do the revised estimates most differ from the original estimates? Why?**

   There are three parts to the smoothing process: forward filtering, backward inferencing, and smoothing. The forward/filter data is the monthly population size of homeless we computed from the Kalman filter. We then performed backward inference and smoothing (August → March) recursively using the data from the following months to get better estimates of the previous months. We set the start of the backward inference process (August) equal to 1271, since taking the initial population size of 1200 and applying all the transitional actions to it for all the months gives us the population size we are expecting to get. The smoothing series begins with the forward/filter data from August, since we do not have further data to begin smoothing. After performing backward inferencing, we performed a weighted mean calculation based on the variances (as weights) of the forward and backward inference results to get the smoothed estimate for the homeless population for all the months.
   In the month of March, we saw the biggest difference in our revised estimate. We expect this to be due to the fact that March was the first month and therefore the least trained estimate. As the time steps increase, the population size variance of the Kalman Filter

system decreases and our Kalman filter starts to get a better understanding of the world to produce better estimates.

2. Your boss would like a projection for the homeless population for the two months after you left (September and October). For your transition model, you should assume the new Mayor will adopt policies that cause the number of homeless to increase.

   The projected population size of homeless people in the month of September and October 2020 is about 1,371 and 1,413 with 8% and 3% increase in population size, respectively, due to the mayor's policies that will increase the number of homeless people in Worcester.

   Please refer to our Excel spreadsheet for more details.

Resources:
[1] https://www.telegram.com/opinion/20180304/as-i-see-it-worcester-must-address-homeless-issue-once-and-for-all
[2] https://www.coalitionforthehomeless.org/state-of-the-homeless-2019/
[3] https://endhomelessness.org/homelessness-in-america/homelessness-statistics/state-of-homelessness-report/massachusetts/
[4] https://www.statista.com/statistics/302328/arrest-rate-in-the-us-2012-by-state/

Part 2. Expectation maximization

Question #1

How did you initialize the cluster centers? Was it random? Did you methodically walk through the search space? Something else?

The clusters were initialized with the use of random numbers. The centroid of each cluster corresponds to a random point of the data set. The variance of each cluster is a random fraction between 70% and 95% of the total variance of each dimension. The assumption of setting an initial variance as a fraction of the total variance is valid because the variance of each cluster should be lower than the variance of the entire data set. Similarly, variance is not set as a random number since it depends on the units of measure of each dimension, which are not known before the application of the expectation maximization algorithm.

Since the assignment allowed the distributions of each dimension to be independent, it was not needed to estimate a covariance matrix between the distributions at each dimension.

Question #2

Explain how you used BIC as a modeling fitting criteria and how you used it to terminate your search.

Since calculating BIC can be rather time intensive, it is only used to compare models with different numbers of clusters. This comparison takes place at an early stage of the iterations, where there are the biggest gains in the calculations of the log likelihood. Thereafter, time is best used to improve the estimation of the parameters of the clusters, which converges with more variance than the log likelihood (as it will be shown in Question #5).

For that matter, BIC was not used to terminate the search, but rather has an intermediate point which decides the optimal number of clusters.

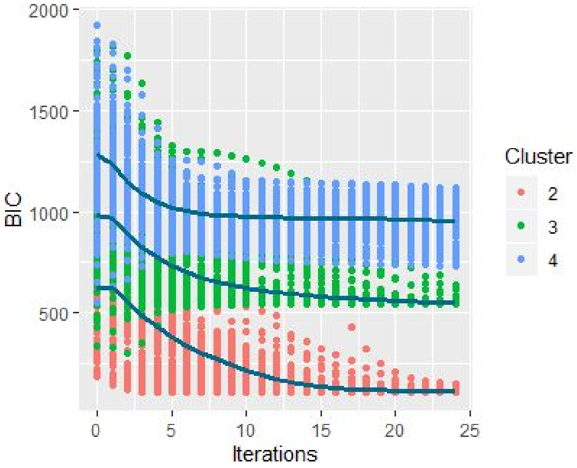
Question #3

How did you trade off: number of iterations, random restarts, and determining the number of clusters? Explain your approach for when to stop doing each of them.

The main functionality of the expectation maximization algorithm is to do a unique random start, and apply all the time left to do more iterations and thus, improve the value of the log likelihood and the estimation of the parameters. When determining the number of clusters, the algorithm will run 5 iterations for two possibilities of clusters (or employ less than 5 seconds of iterations if it can not fulfill 5 under that time), and then choose the one with the lowest BIC to employ the time left to proceed with the iterations. Each of these considerations will be explained next:

Number of iterations and Random Restarts:

Using the sample file that was provided for the assignment, 100 random restarts were run, with 2, 3 and 4 clusters, making a total of 300 experiments. Each start had 25 iterations to improve the value of BIC and the parameters. The results of the BIC for all these values are shown below:



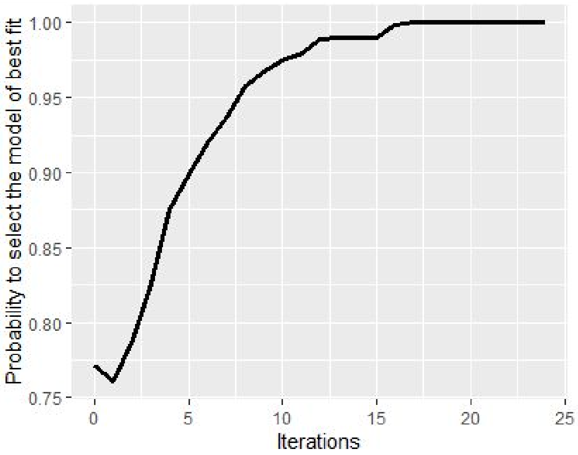
The main conclusion of the previous graph is that regardless of the random starting points, all the values of the BIC converge to a value determined by number of clusters (a similar behavior is shown at Question #5 for the parameters). Even if expectation maximization is a type of a hill climbing algorithm, these experiments do not support the idea to do random restarts in the hope of surpassing a local maximum. This realization is very important because time is considered a scarce resource for simulations at high dimensions and a large number of clusters. As such, it is more valuable to do just one random start and use the remaining time to improve the model fit.

Determining the number of clusters:

Analyzing the last iterations, one can affirm that 2 is the number of clusters of best fit. Despite this data being created by three clusters, the BIC criterion is lower for 2, and this is ultimately the information that will decide how many clusters are classified in the data.

Model selection confidence is not the same at the different stages of the iterations. Once the log likelihood values have converged, it is easy to identify which number of clusters best suits the data. Nevertheless there is more uncertainty at the initial iterations, since they have parameters that are closer to the random restart values, which do not consider the distribution of the data.

Using the previous experiment with the provided data set, there are 300 hundred points for each iteration; 100 BIC values for each cluster. To approximate the probability to select the model with 2 clusters as the model of best fit, a random sample of three points was taken, each representing one of the clusters. Then, the model with the lowest BIC was selected. The next graph shows an approximation to correctly choose the model with 2 clusters, from 15,000 random samples at each iteration.



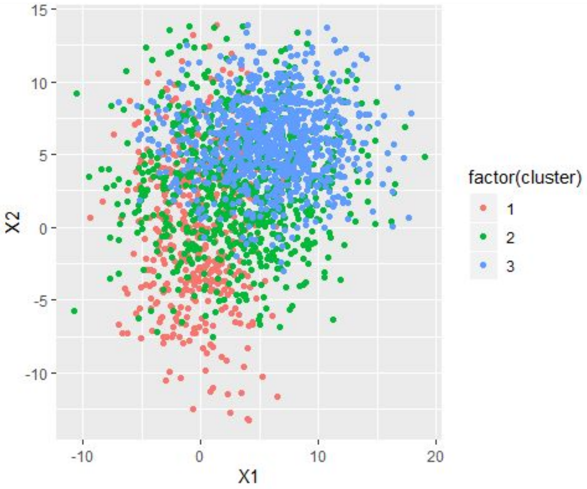
As it can be seen from the graph, the probability to select the model with 2 clusters increases with the number of iterations. However, this probability is already near to 90% for the fifth iteration. This means that the model of best fit can be identified at the early stages of the iterations with a high confidence, leaving the rest of the time to improve the quality of the log likelihood and the means and variance from the centroids.

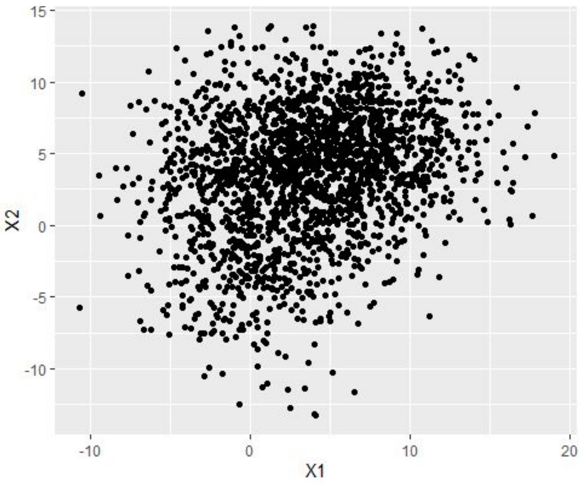
Finally, there is an initial approximation of the number of clusters based on a gaussian kernel of the data. Once this kernel is obtained from one of these dimensions, it is identified how many peaks there are at the kernel distribution, which would signal the number of clusters. The procedure is repeated for all dimensions and the initial approximation of the number of clusters of the data equals the maximum number of peaks at the gaussian kernels for one of the dimensions. Since this approach might underestimate the number of clusters because they could be close to each other, the BIC is evaluated for this approximation and for one extra centroid with one additional cluster.

Question #4

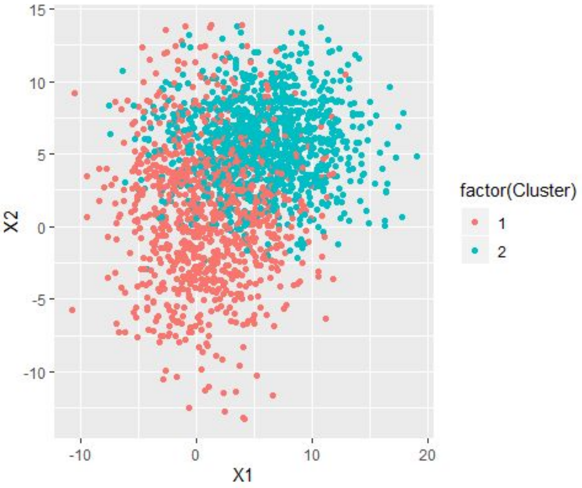
Create a data file whose clusters are not easily separable. How does EM perform? Specifically, how accurate is it in determining the correct number of clusters? If given the correct number of clusters, does it find the correct means and variances of the clusters? Does it assign points to the correct cluster? Answer the same questions for the provided sample data file.

The data with clusters that are not easily separable are shown in the next graphs, which represent the information with and without the cluster labels:



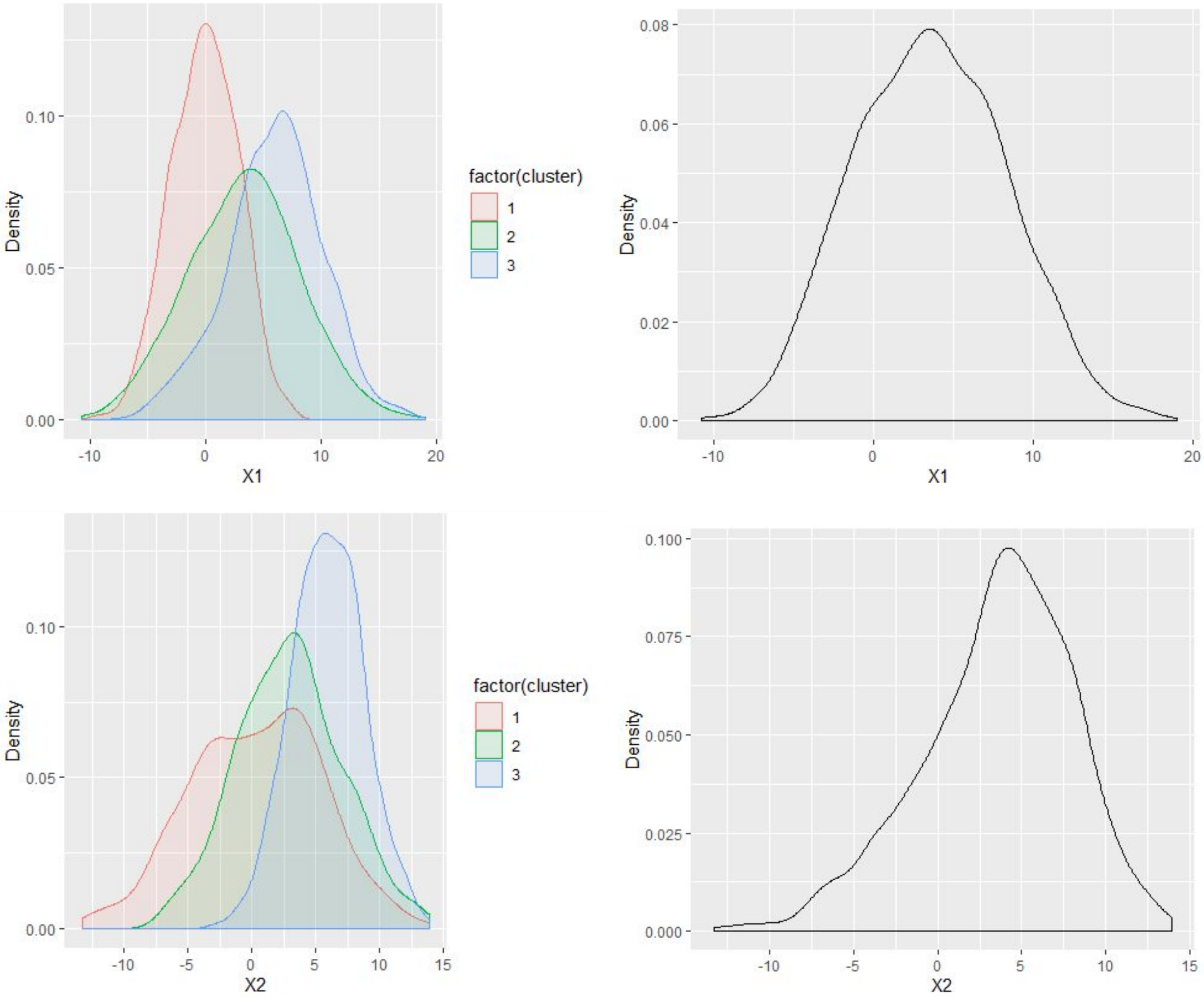


The performance of the algorithm was able to identify two clusters, that are shown below:



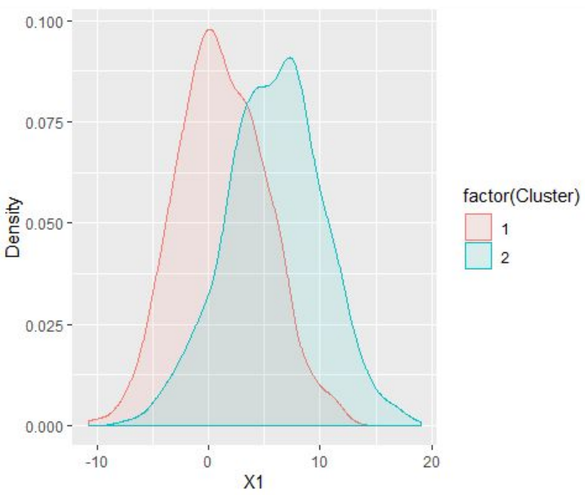
The expectation maximization algorithm does not identify the number of clusters in a perfect manner. The data was generated by three clusters, but EM only identifies two. Despite this result, it is capable of discerning the clusters that are positioned further apart.

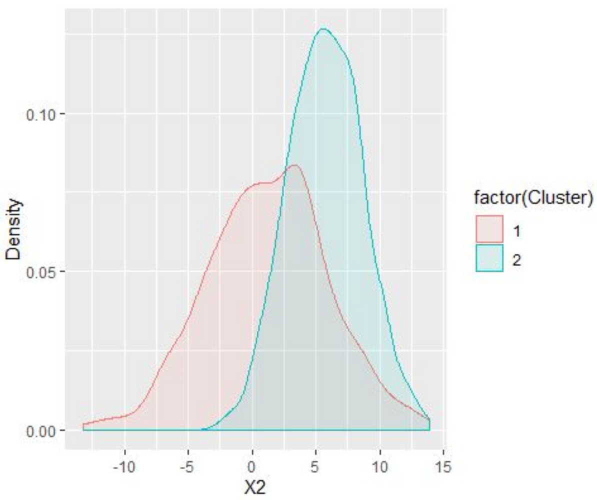
However, there might be an underlying reason the expectation maximization algorithm would have problems with close clusters. If independent clusters are not easily separable, they are likely to form a bigger normal distribution with a bigger variance. This can be appreciated in the histograms for the initial data, where the aggregate distribution of the information looks similar to just one normal gaussian.



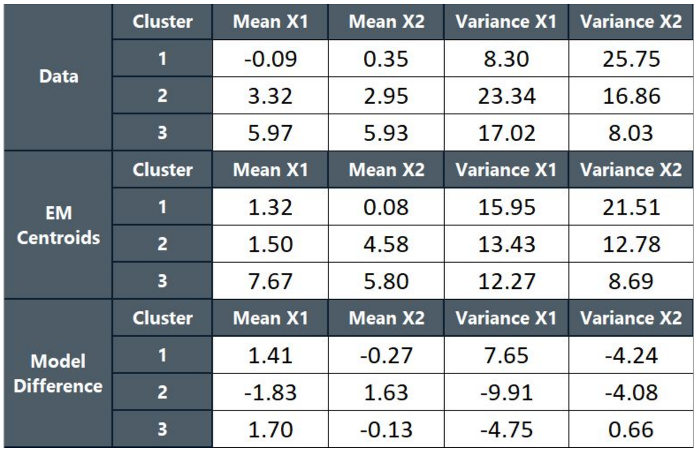
Given this behavior with clusters that are close to each other, it is expected that the algorithm could underestimate the number of clusters in the data. But without the a priori information of knowing that there are three clusters instead of two, it is complex to argue which is the number of clusters that best describes the data.

The next couple of graphs shows that the clusters that are more separated from one other are capturing the variance and observations of the cluster that lies in the middle. This distribution of the data of the middle cluster is expected since the clusters are not easily separable.

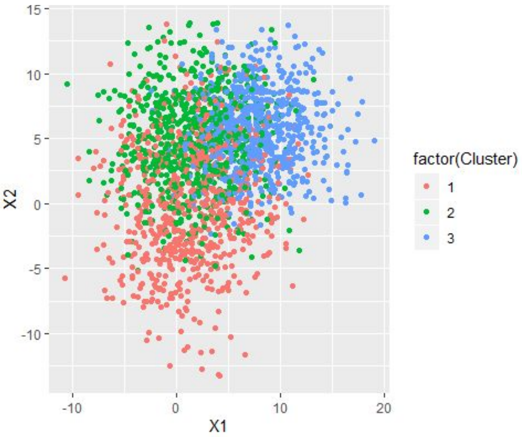




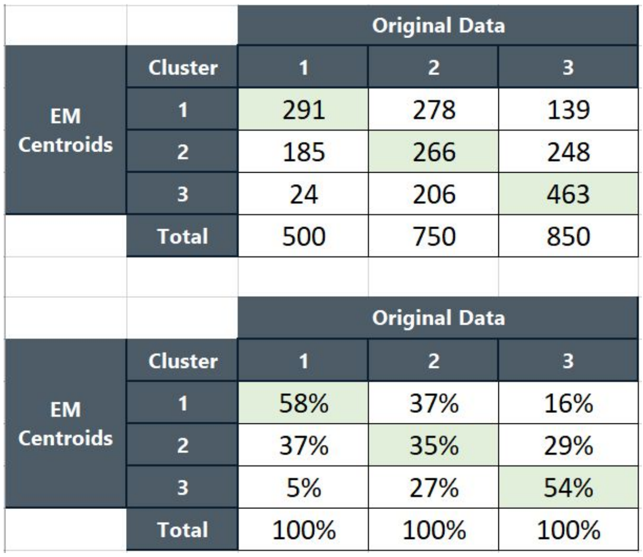
If given the correct number of clusters, the calculation of the center of the centroids returns a good approximation. As it can be observed in the next graph, the means of the centroids are not further than 2 units for all the dimensions and clusters. It would also be expected that with more time than the 10 seconds, the algorithm would arrive at a better result. Nevertheless, the variance of the true centroids are notably different than those estimated by the algorithm. As will be discussed in the following question, the mean values of the centroids converge more quickly than the variances.



	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
Data	1	-0.09	0.35	8.30	25.75
	2	3.32	2.95	23.34	16.86
	3	5.97	5.93	17.02	8.03
	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
EM Centroids	1	1.32	0.08	15.95	21.51
	2	1.50	4.58	13.43	12.78
	3	7.67	5.80	12.27	8.69
	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
Model Difference	1	1.41	-0.27	7.65	-4.24
	2	-1.83	1.63	-9.91	-4.08
	3	1.70	-0.13	-4.75	0.66



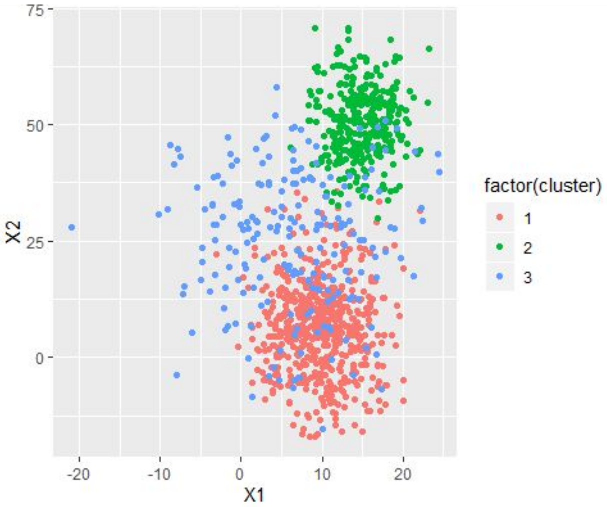
The accuracy of the algorithm is represented in the next matrices. As an overall result, there is a correct accuracy in the identification of the clusters of 50%. Certainly this is not a good result, but one has to acknowledge that clusters are really close together. In fact, the biggest proportion of the mistakes in the classification are present or related with cluster 2, that is positioned between 1 and 3. One can observe that there is a small percentage of misclassification between clusters between 1 and 3 (that equal 5% and 16%). Similarly, the biggest proportion of the wrong assignation of clusters is at 2.

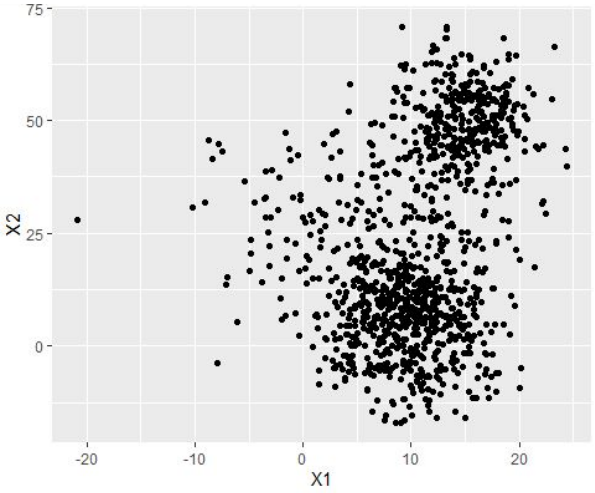


		Original Data		
	Cluster	1	2	3
EM Centroids	1	291	278	139
	2	185	266	248
	3	24	206	463
	Total	500	750	850

		Original Data		
	Cluster	1	2	3
EM Centroids	1	58%	37%	16%
	2	37%	35%	29%
	3	5%	27%	54%
	Total	100%	100%	100%

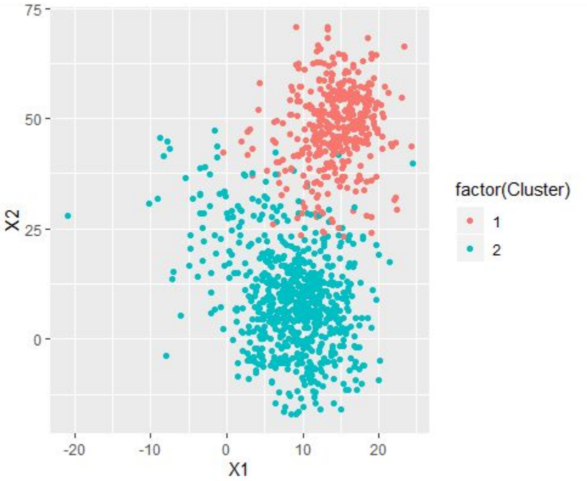
Since the provided data has clusters that are more dispersed than the previous example, it can be expected that the results are going to be better for this situation. To begin with, the next graphs visualize the data with and without the correct classification.



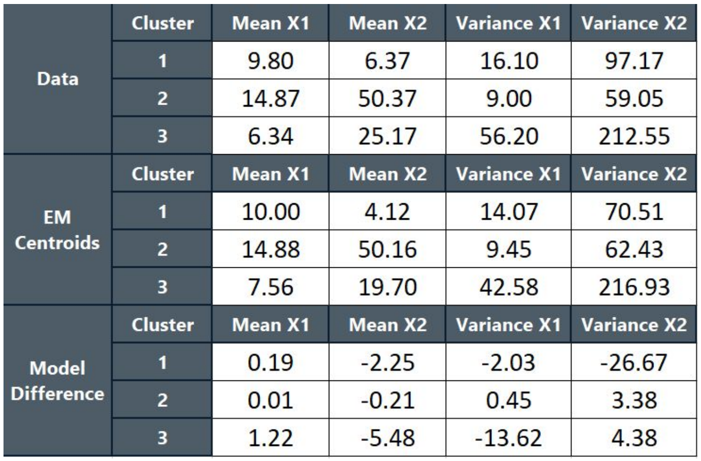


In this case, it can be observed once more that there is a cluster in the middle of two better identified clusters that lie in the extremes of the graph. Moreover, the cluster that is positioned at the middle has more variance than the rest, which will make it more difficult for the algorithm to correctly identify the number of clusters in the data.

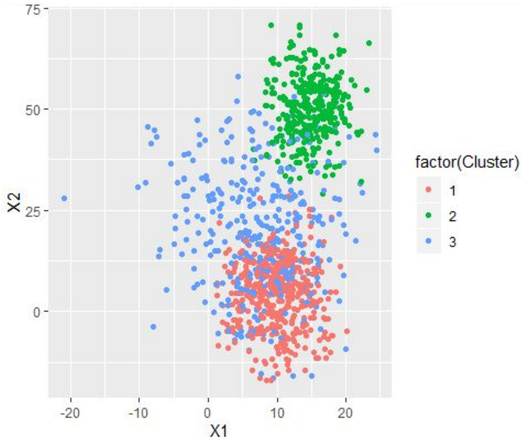
As it was expected, the algorithm outputs just two clusters for the data. Nevertheless, one has to remember (from Question 3) that this is the correct answer, since this is the number of clusters that minimizes the BIC criterion for this data.



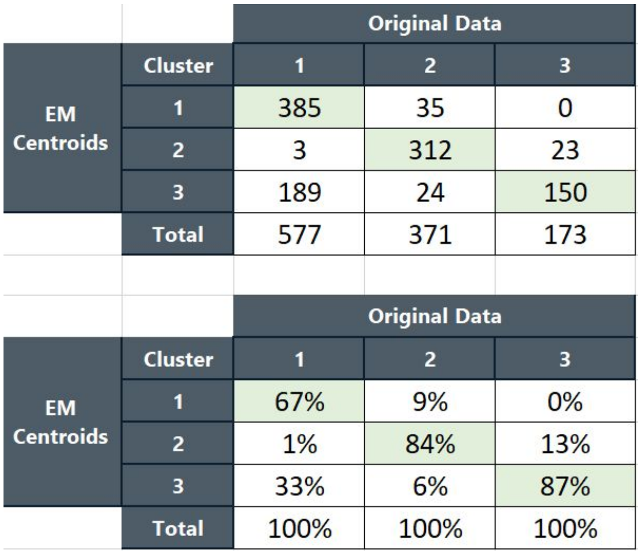
If given the correct number of clusters, the algorithm delivers a good approximation of the parameters. In this case, the good results are also extended for the variance. Cluster 2 is the one that is further apart and more clearly differentiated from the others. In consequence, the estimation of their parameters is closer to the real values of the data. There is a bigger difference for clusters 1 and 3 because they share a relevant proportion of values alongside the second dimension, pictured as X2 in the next graph. Nevertheless, the results are still good approximations of the real values of the distribution.



	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
Data	1	9.80	6.37	16.10	97.17
	2	14.87	50.37	9.00	59.05
	3	6.34	25.17	56.20	212.55
EM Centroids	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
	1	10.00	4.12	14.07	70.51
	2	14.88	50.16	9.45	62.43
	3	7.56	19.70	42.58	216.93
Model Difference	Cluster	Mean X1	Mean X2	Variance X1	Variance X2
	1	0.19	-2.25	-2.03	-26.67
	2	0.01	-0.21	0.45	3.38
	3	1.22	-5.48	-13.62	4.38



The last graph of this section shows the correct identification of points to their respective clusters. In contrast with the previous example, the algorithm outputs a correct classification of 75% of the data. The bigger problem of this result is the misclassification between clusters that should belong to one but were assigned to 3. This is explained because this pair of clusters share a distribution of the data in the dimension represented by X2.



		Original Data		
	Cluster	1	2	3
EM Centroids	1	385	35	0
	2	3	312	23
	3	189	24	150
	Total	577	371	173

		Original Data		
	Cluster	1	2	3
EM Centroids	1	67%	9%	0%
	2	1%	84%	13%
	3	33%	6%	87%
	Total	100%	100%	100%

In summary, the output of the expectation maximization algorithm provided a good approximation of the values of the parameters and a good classification of the data point. However, the quality of this output is restricted by time, because several iterations were needed to converge to a value of the means, variance and log likelihood, and also by the inherit variance and proximity of the clusters, when either of this characteristic increases, it is also more difficult to have good results.

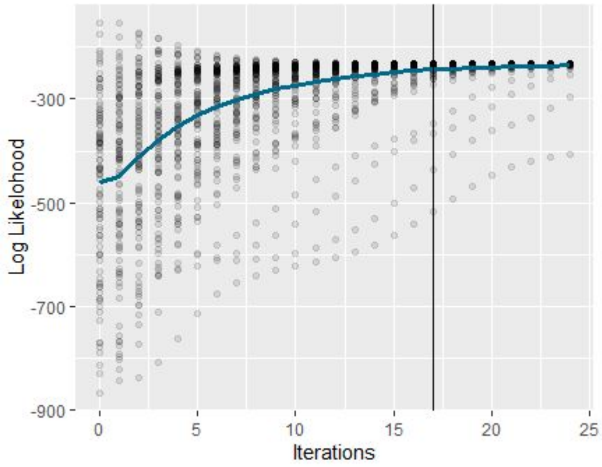
Question #5

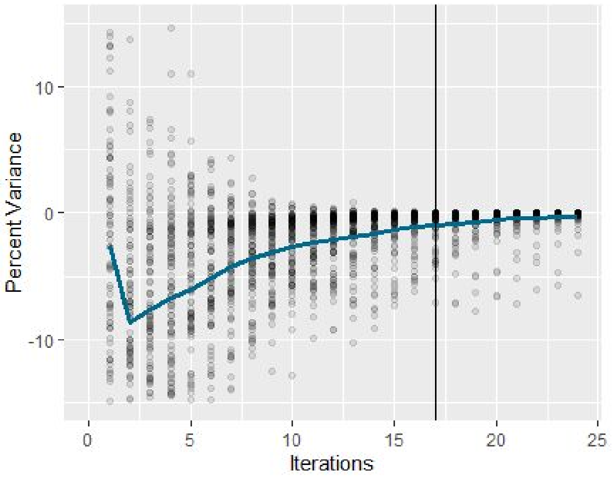
Sketch log-likelihood vs. # of iterations for a data set. Run it for longer than your typical termination criteria, and mark where the algorithm would normally stop on the graph. Provide the parameter estimates for where your program would normally stop, and what it would find if it kept going. Does convergence of log-likelihood correspond to convergence of model parameter estimates?

Using the provided sample data set, and limiting the number of iterations to the time of 10 seconds of execution, the expectation maximization algorithm usually stops at the 17th iteration. To evaluate how the log likelihood changes over the iterations, 100 simulations were run with random initial values for the clusters' means and variances.

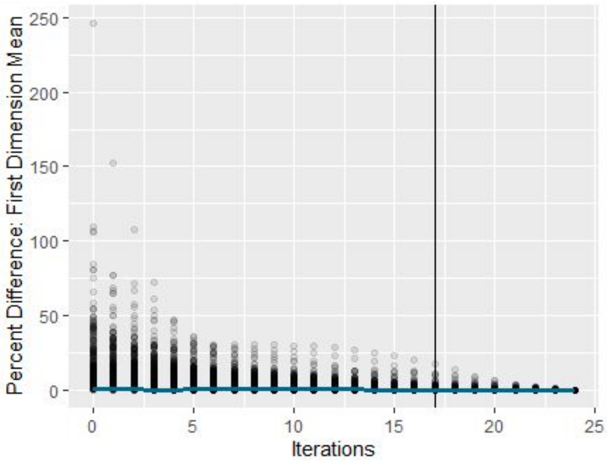
The following graphs show the evolution of the value of the log likelihood and the percent variance of the log likelihood. The left graph shows that the log likelihood converges to a value around -235. The path to this value has a notable variance, especially at the first iterations. Similarly, the biggest gains in the log likelihood are also present at the first iterations. That means that the expectation maximization needs over 20 iterations to arrive at the best fit for this sample file, but that the higher improvements for the model are present at the first iterations.

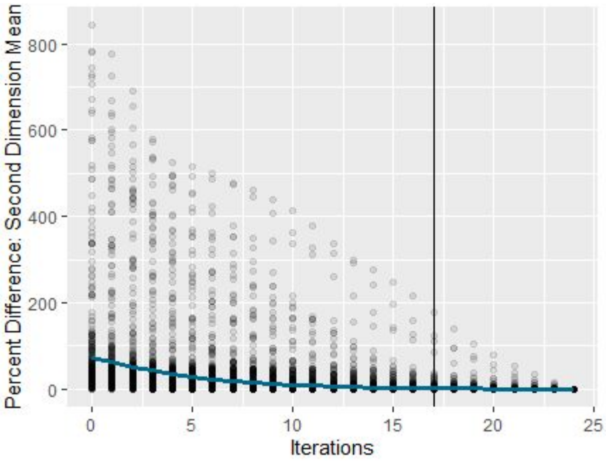
The graph on the right provides an approximation of the improvements and convergence of the values for the log likelihood. It displays the percent variance in correspondence with the lagged value of the previous iteration. From the average of the 100 simulations, there is a notable decrease of the likelihood function until the second iteration, which is rounded to 9%. After this step, each iteration reduces the likelihood function in a lower magnitude: the log likelihood reduces less than 3% from the 10th iteration, and lower than 1% from the 17th iteration.

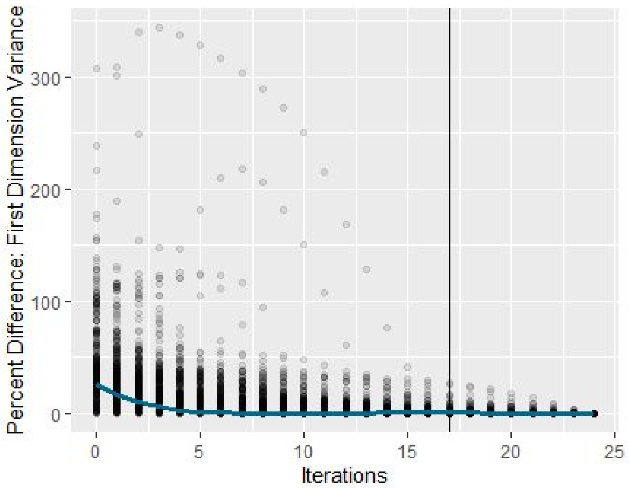


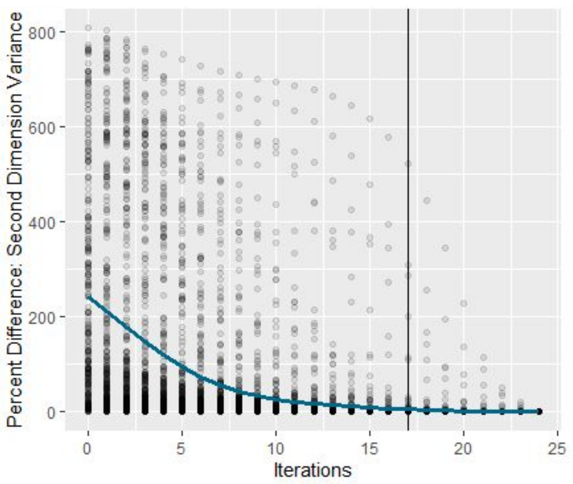


The following graphs present a similar set of figures, but they present the evolution of the parameters through the iterations. Each ordinate value represents the absolute value of how far is the parameter from the final value. Despite the four graphs showing a certain kind of convergence when iterations are above 17, all of the data points hold a significant amount of variance. The variance is even more drastic at the second dimension of the data, which has data that is distributed in a longer range.









The comparison between the evolution of the log likelihood and the values of the parameters under this test file shows that there is a correspondence in the converge when several iterations were made. Nevertheless, the convergence of the log likelihood is much smoother than the one of the parameters, which holds a large variance, specially for more sparse clusters.