GTID: jliu727

Assignment 3: Unsupervised Learning and Dimensionality Reduction

Data:

College Ranking Score:

This is the dataset I used from Assignment 1. This dataset includes university ranking data from year 2012-2015 from The Center for World University Rankings is a listing that comes from Saudi Arabia founded in 2012. The raw data includes in total 2200 instances with 14 attributes. The attributes include information such as education and research power in numerical format. I used score as the output so I eliminated the columns of 'world_rank', 'national_rank'. Also, 'institution', 'country' and 'year' do not seem to be relevant to the final score and the former two would produce a lot of features if I were to create dummy variables for them so I eliminated them. I also eliminated 'broad_impact' as it lacks data. In this case, I have 7 features left. This dataset is interesting because its output is 'score' which is continuous without clear classification, I would like to see how it behaves compared to the datasets with its own classification in clustering process.

Wine Quality:

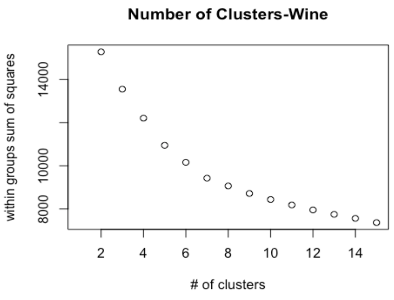
This is a new dataset containing information on different wine such as acidity, sugar, density, pH etc. Then a score for quality is produced for each instance. The dataset has 11 features. This dataset is interesting because it has a limited number of quality differences. I would like to see how clustering and dimensionality reduction does with the existence of its original classification. I have chosen this dataset over my previous dataset on mushrooms because the number of features for the mushroom dataset is too large (> 100) and could cause processing time concern so I changed to this dataset which is also a classification dataset but with limited number of features.

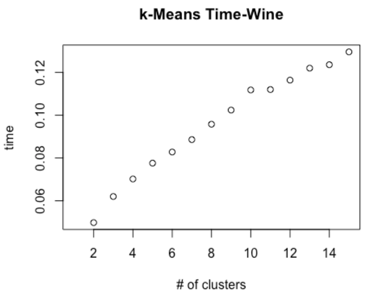
Clustering:

k-means:

In k-means, we randomly choose k centroids and group their nearest points to them, then we recalculate the centroid and repeat the previous process until the the centroid is stable. In this method, a point can only be clustered into one cluster. For the k value, sometimes, it makes sense to estimate a k value based on how many classification labels the data has. In general, we would use the elbow method to determine what is the best k value for the clustering. We do not want the k value to be too large because that would be meaningless as we think about the extreme case where each data point is a cluster. At the same time, we do not want to the k value to be too small because that would cause the danger to not reveal enough information.

Wine Data:

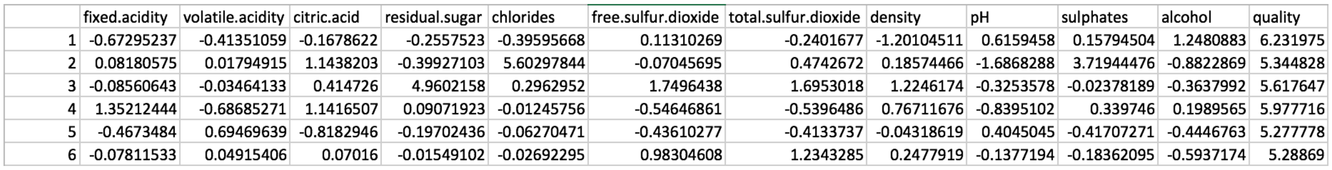




GTID: jliu727

As the wine dataset is already a classification dataset with 6 different quality level, in practice, it would make sense to use k=6. Interestingly, we can see the graph that sum of squared within clusters drop rapidly and starts to level around k = 7 which roughly matches up with our intuition to use k = 6. We can also see that the time to run k-means method is mostly linearly proportional to the # of clusters which makes sense as each round it is recalculating the center and distance to try regrouping data points which takes approx. the same amount of time each iteration. Here we use k=6 so it is easier to compare with the actual classification later.

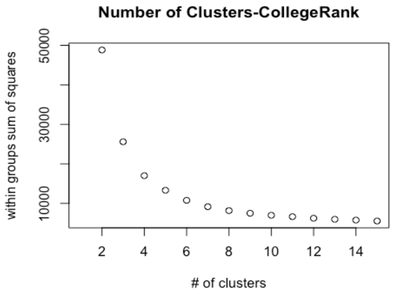
Result: k-means has divided the data into 6 clusters of size 319, 29, 34, 359, 522, 336.
When looking at the cluster mean for each of the features below:

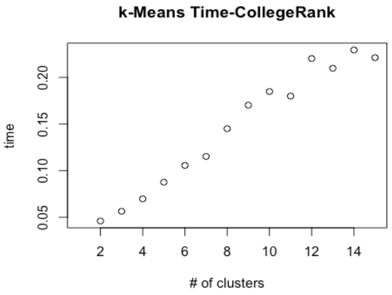


	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1	-0.67295237	-0.41351059	-0.1678622	-0.2557523	-0.39595668	0.11310269	-0.2401677	-1.20104511	0.6159458	0.15794504	1.2480883	6.231975
2	0.08180575	0.01794915	1.1438203	-0.39927103	5.60297844	-0.07045695	0.4742672	0.18574466	-1.6868288	3.71944476	-0.8822869	5.344828
3	-0.08560643	-0.03464133	0.414726	4.9602158	0.2962952	1.7496438	1.6953018	1.2246174	-0.3253578	-0.02378189	-0.3637992	5.617647
4	1.35212444	-0.68685271	1.1416507	0.09071923	-0.01245756	-0.54646861	-0.5396486	0.76711676	-0.8395102	0.339746	0.1989565	5.977716
5	-0.4673484	0.69469639	-0.8182946	-0.19702436	-0.06270471	-0.43610277	-0.4133737	-0.04318619	0.4045045	-0.41707271	-0.4446763	5.277778
6	-0.07811533	0.04915406	0.07016	-0.01549102	-0.02692295	0.98304608	1.2343285	0.2477919	-0.1377194	-0.18362095	-0.5937174	5.28869

We can see that the clusters reveal little information about the original classification: quality. Except for cluster 1 which had a slightly higher quality score, all the other clusters are mostly around score 5. The reason that k-means is not working well may be that quality depends on too many variables and these variables can be highly independent from each other. We can not say definitely that high density indicates high quality and low density means low quality. Wine with similar quality can be physically far away from each other as points, indicating Euclidean distance may not be a good measure of similarity in wine quality.

College Ranking Data:

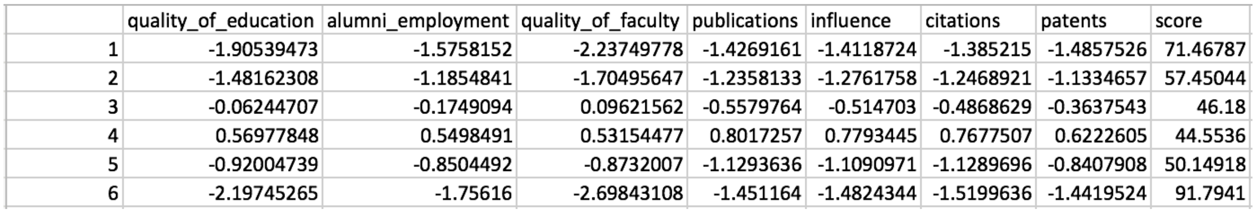




This dataset is different from the wine dataset as it does not have classification label. It only has a continuous output 'score'. Therefore, there is no good way for us to estimate the best k number to use by looking at the data. We can see from the graph above that around k = 6, the rapid decrease of sum of squares starts to level. Therefore, we are going to use k = 6 in this case.

Result: k-means has divided the data into 6 clusters of sizes 47, 114, 570, 1138, 292 and 39.
When looking at the cluster mean for each of the feature below:

GTID: jliu727



	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	patents	score
1	-1.90539473	-1.5758152	-2.23749778	-1.4269161	-1.4118724	-1.385215	-1.4857526	71.46787
2	-1.48162308	-1.1854841	-1.70495647	-1.2358133	-1.2761758	-1.2468921	-1.1334657	57.45044
3	-0.06244707	-0.1749094	0.09621562	-0.5579764	-0.514703	-0.4868629	-0.3637543	46.18
4	0.56977848	0.5498491	0.53154477	0.8017257	0.7793445	0.7677507	0.6222605	44.5536
5	-0.92004739	-0.8504492	-0.8732007	-1.1293636	-1.1090971	-1.1289696	-0.8407908	50.14918
6	-2.19745265	-1.75616	-2.69843108	-1.451164	-1.4824344	-1.5199636	-1.4419524	91.7941

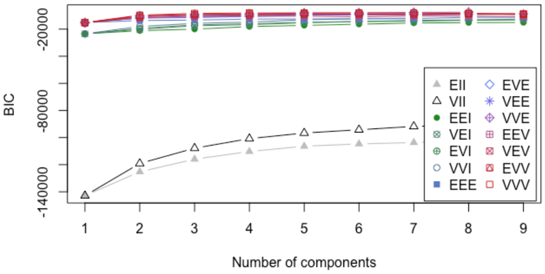
We can see that it is pretty well divided in scores. Half of the clusters have average below 55 which makes sense as around that much data does have the score below 55. K-means is working fine here maybe because the score is highly relevant to Euclidean distance between each points in space. It makes sense intuitively as all features here are monotonic to the score. Increase in one feature would mean the increase in score so overall we can imagine different ranges of score can be nicely divided in space.

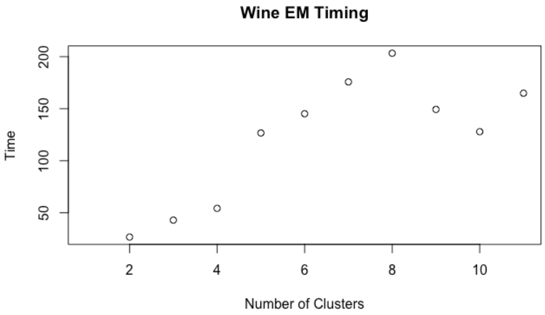
Expectation Maximization:

Wine Data:

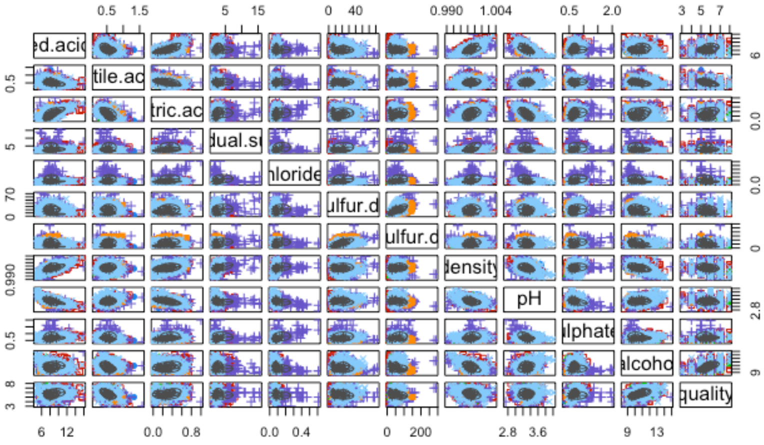
The BIC graph levels off at around k = 6, therefore, we use the same k value as before.

We can see here that the time to execute EM is much longer than k-means which is expected as it needs to create Gaussian distributions and calculate the probability of each points existing in those distributions which is way more complex.





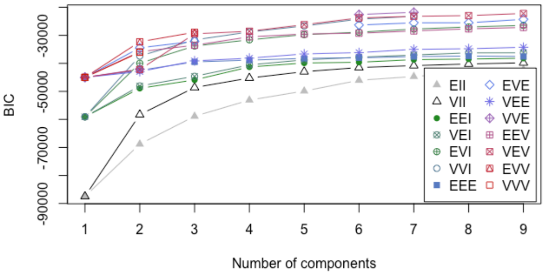
Result:

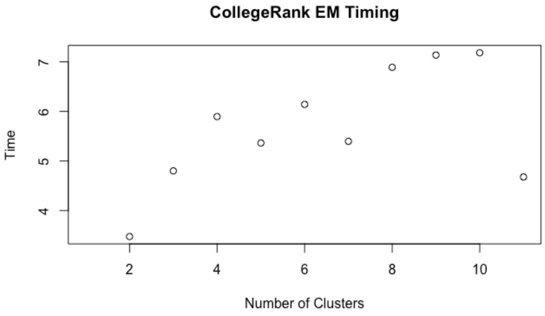


GTID: jliu727

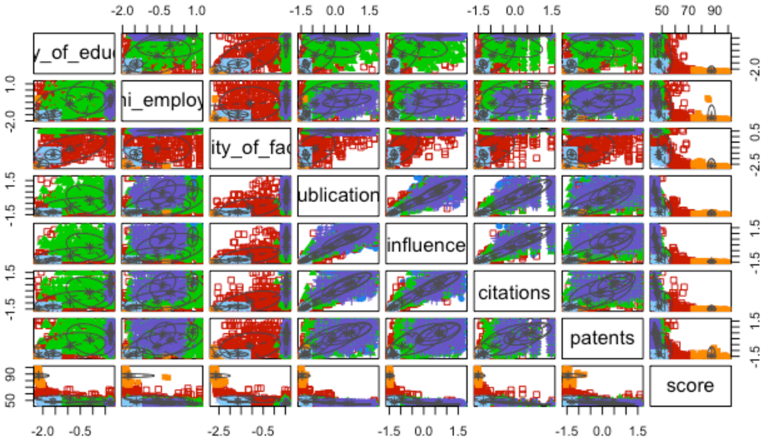
We can see that there are too any features that it is almost impossible to read from the graph how well the clustering is done but we can see the clusters are mostly still tangled together, not dividing very well. Feature reduction may help in better expressing how the clustering is doing.

College Ranking Data:
The BIC graph levels off around k=6 or 7 so I am going to use k = 6 as in k-means.
We can see the execution time for EM still takes much longer than k-means.





Result:



Here, we can see score is a big indicator in clustering as it divides data points pretty well in the last row and column.

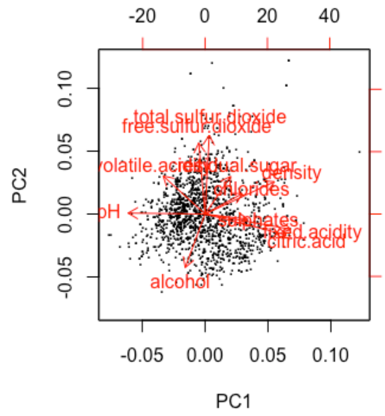
Dimensionality Reduction:

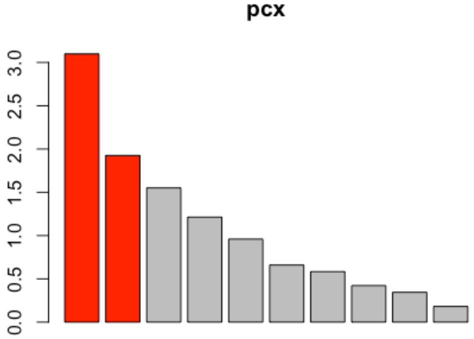
PCA:

Principal Component Analysis is a global algorithm that looks for components with large eigenvalues. It eliminates components with small eigenvalues and map the data to a new lower dimensional space to reduce dimensionality.

GTID: jliu727

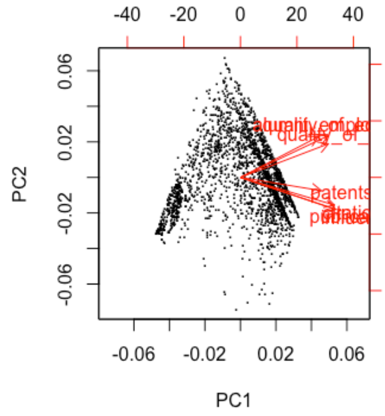
Wine Data:

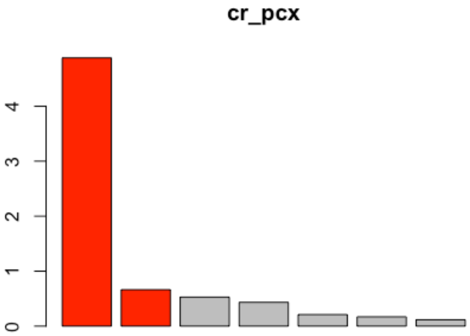




We can see the data points in the space of the first two principal components on the left and the variance of principal components on the right. Looking more into pcx, it actually created 11 principal components. We can see that after 6 principal components, the variance became pretty low. Eigenvalue decreases as variance decreases so we can ignore the last few principal components.

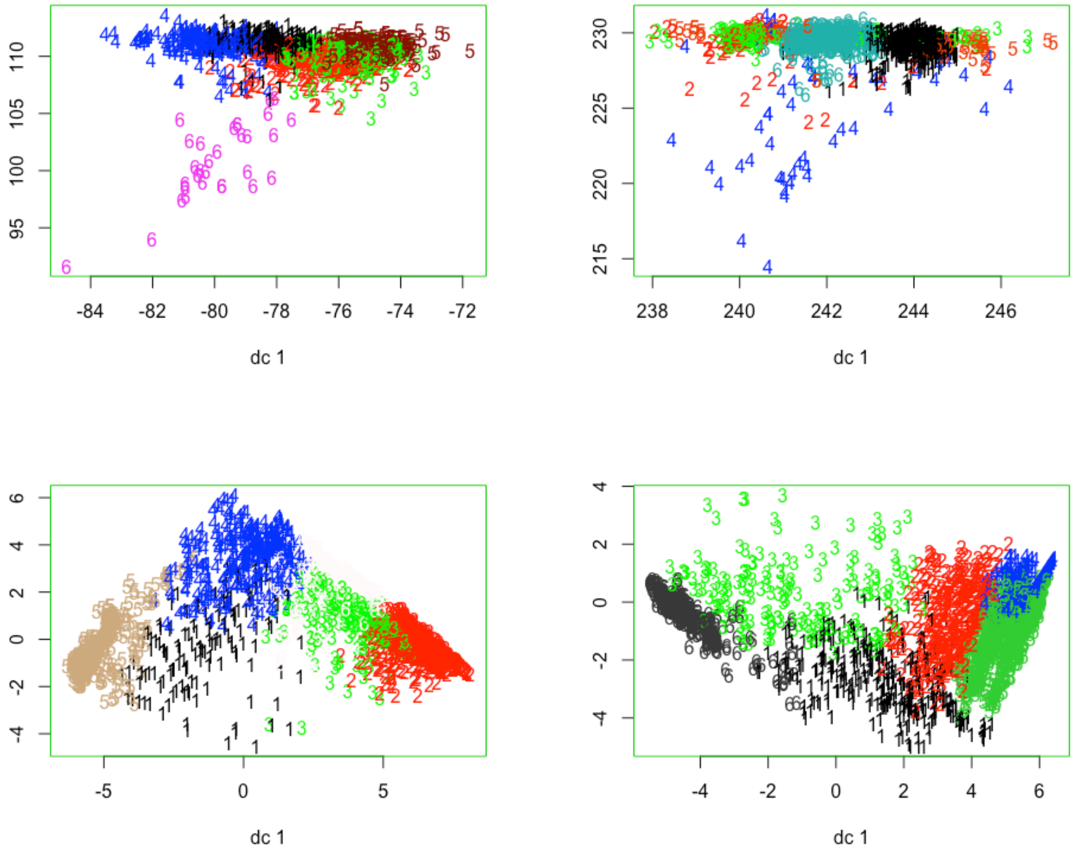
College Ranking Data:





We can see in this graph that the 1^{st} principal component contains the most information as its variance is much higher than the rest.

We apply clustering again on the dataset after PCA dimensionality reduction:

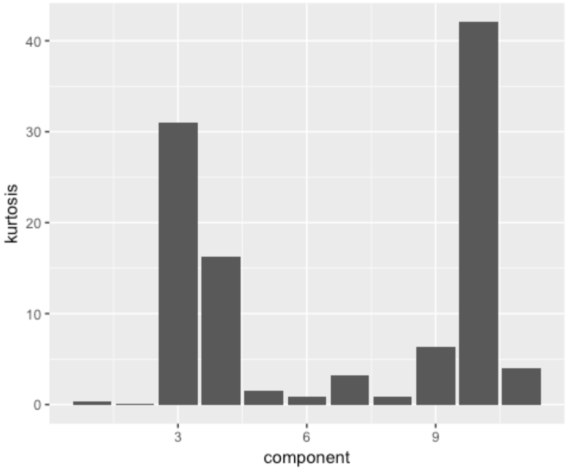


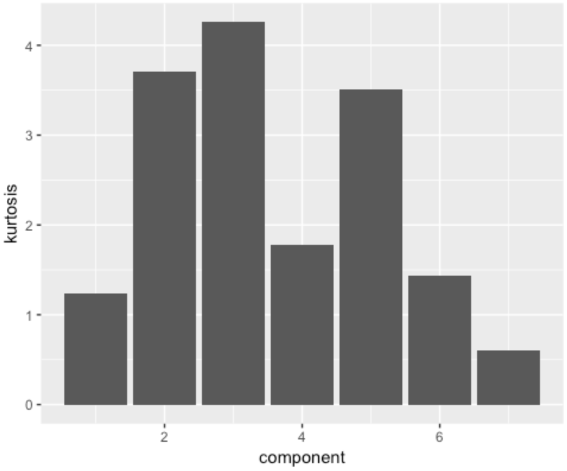
Graph order: upperleft: k-means + wine, upperright: EM + wine, lowerleft: k-means + college ranking, lowerright: EM + college ranking. Same order for sections below.

We can see for data wine quality with addition of PCA, EM is doing slightly better then alone. It could be because PCA has reduced the the dimensionality it has became less complicated for EM to construct clusters. Data college ranking looks good in clustering with both k-means and EM possibly because of its simple data structure.

ICA:

Independent Component Analysis reduce dimensionality by finding components that maximizes independence. These components do not have order as PCA and we can find the best components by looking into kurtosis. We want to take the components with large kurtosis value.

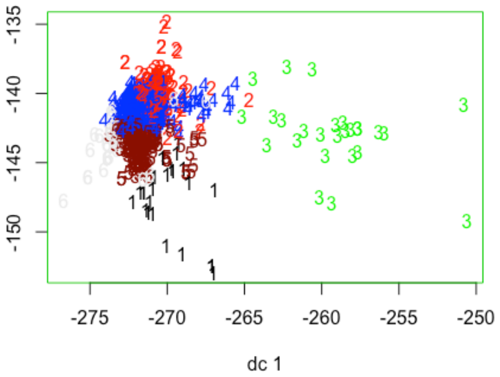


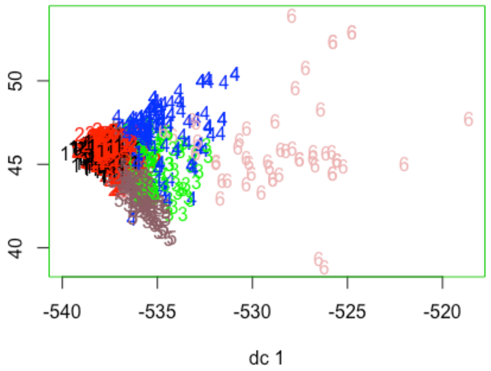


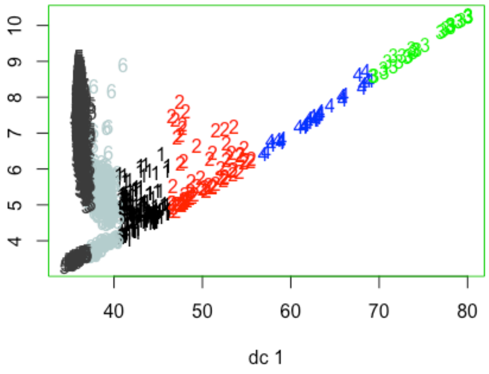
We can see in the wine data on the left, there are three components with large kurtosis, indicating that these are the components containing the most information for reconstruction. We are going to use only three components in this case.

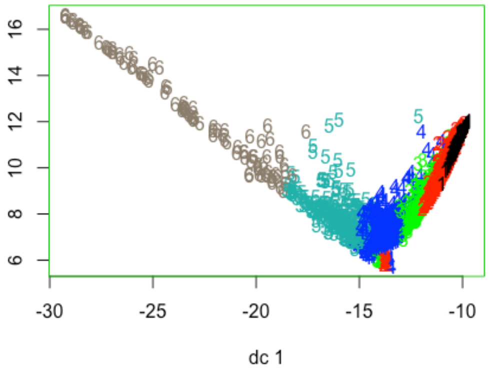
For College Ranking on the right, although not as dramatic as the graph above, it also shows three components with larger kurtosis, containing majority of the information. We are going to use 3-4 components in this case

We apply clustering again on the dataset after ICA dimensionality reduction:





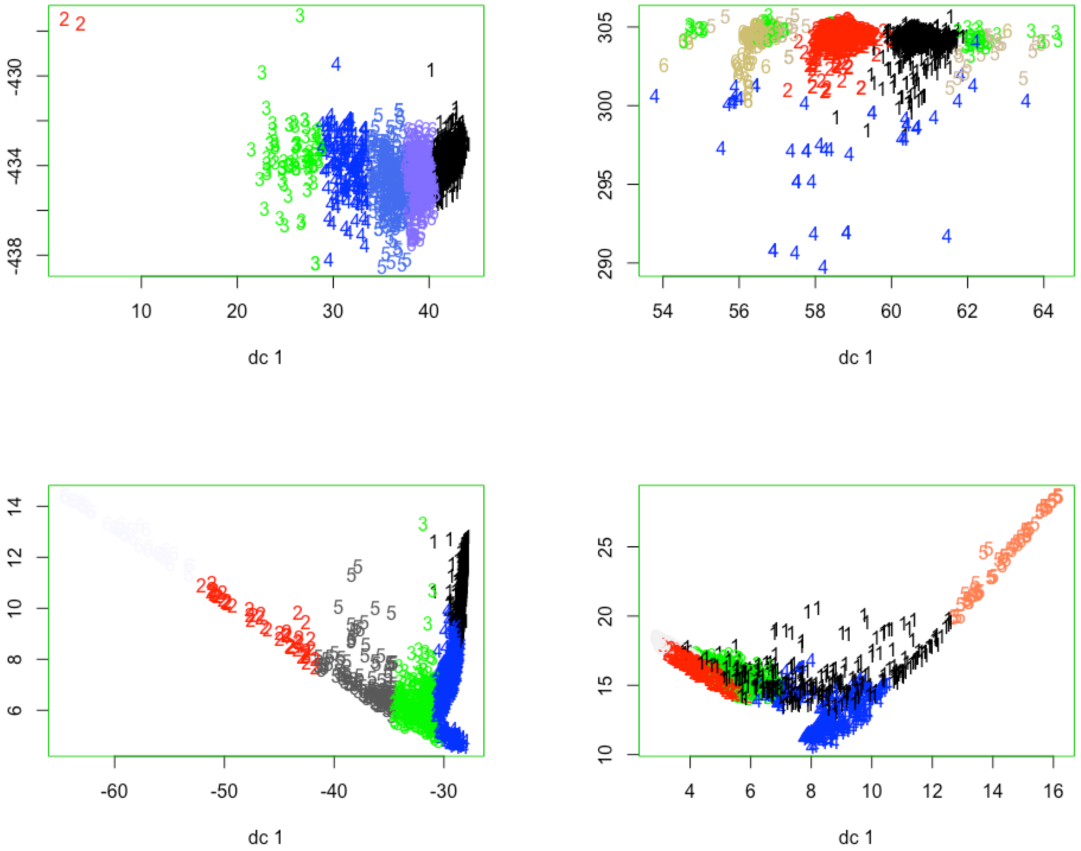




We can see that clustering for data wine quality after being processed by ICA performs pretty bad. The reason could be that ICA is not so good at clustering data with features that are likely to be dependent on each other. For wine, I assume that properties would inter-connect as many features are affected by same factors such as skills, geography etc.

Randomized Projections:
Randomized Projections is similar to PCA except that it picks random components and generated random directions. This method is generally very quick because it does not need to calculate the find the 1st principal components while still yielding good result. Here we set k=6 same what we set for PCA.

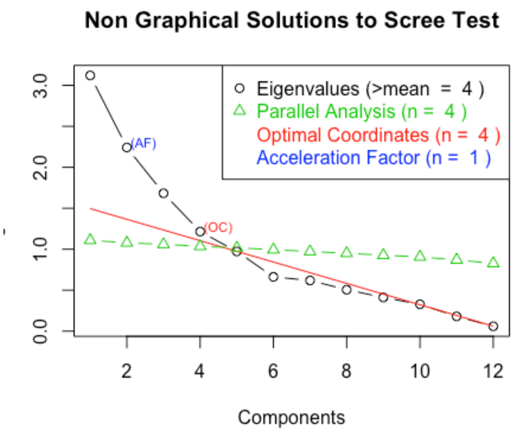
We apply clustering on the dataset after RP dimensionality reduction:

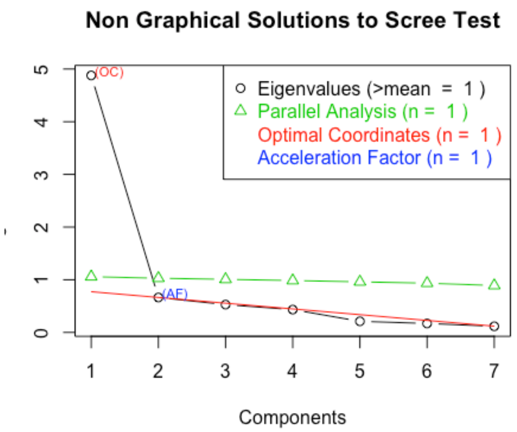


We can see interestingly that after randomized projections, k-means for data wine quality is more clearly divided and it seems that one of the dimension is mainly responsible for the division and other components are not having a large impact. It could be that among randomly chosen components, most of them happen to have very small eigenvalues, therefore, having little impact.

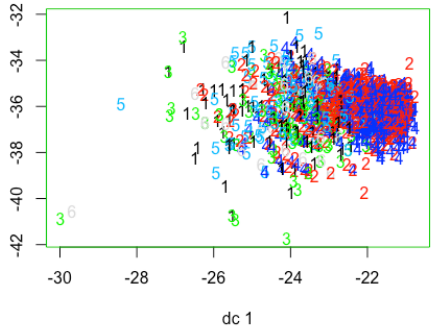
GTID: jliu727

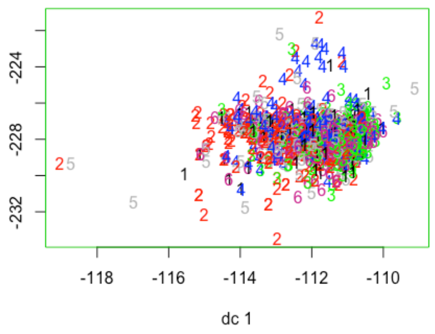
Factor Analysis:
Factor Analysis also aims to find independent hidden components. The observed features are linear combinations of the potential components. It is very similar to ICA except that Factor analysis assumes that the observed variable is linear combination of Gaussian components whereas ICA does not have this Gaussian assumption.

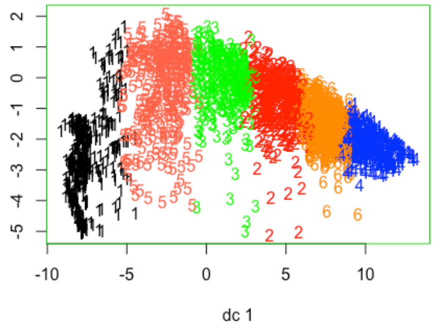


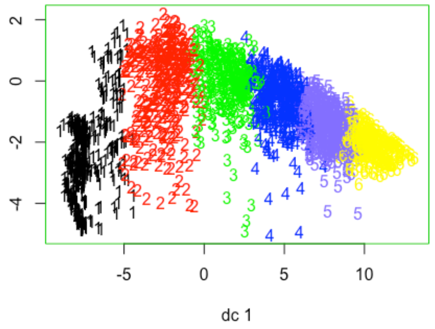


We can see that the optimal components here are 4 for wine data on the left and 1 for college ranking data on the right.
We apply clustering on the dataset after FA dimensionality reduction:









As expected, we can see factor analysis is doing poorly on wine quality data. It could be because of the same reason as for ICA. Also, it could be that it is assuming Gaussian components but it could be that not all variables are good to be modeled as combination of Gaussian components. It is not doing too poorly compared to ICA on college ranking data because there is only one dimension left for the data.

Neural Network using Dimensionality Reduction:
Below is the accuracy reached by Neural Network applied after dimensionality reduction:

Neural Network: 0.8945
PCA + Neural Network: 0.7355
ICA + Neural Network: 0.3682
RP + Neural Network: 0.5536
FA + Neural Network: 0.5972727

We can see that none of the dimensionality reduction outperforms simply having neural network on the original data. I think it is because the original data is simple enough and by reducing dimensionality, we have lost information in the process and thus could not accurately predict the result. We can also see that ICA performs the worst which makes perfect sense as it did not provide any useful information in its clustering index and it could serve as a distraction to the classification process. As for execution time, all of them executed pretty fast, comparable of simply running neural network. The only exception is that Randomized projections ran faster which makes sense as it does not calculate its components but just randomly choosing them.

Neural Network using Clustering:
Below is the accuracy reached by Neural Network applied after clustering:

Neural Network: 0.8945
k-means: 0.9163636
EM: 0.4727273

We can see that k-means clustering actually increased accuracy slightly than the raw data. This matches what we saw in the k-means section that the clusters well represent the increase in average score. However, EM is not performing well which can also be understood from EM section, as the cluster looks fairly bad where score is not present. As for speed, k-means has been much faster than EM as discussed above in the 1st section. It is not very apparent in this section as the dataset size is small.