STA5077Z Assignment

Blake Cuningham CNNBLA001

Contents

1 Project 1: Leukemia dataset

1.1 Introduction

The Leukemia data-set used in this project provides an interesting challenge - the number of variables far exceeds the number of observations. For this reason, there is a need to employ dimensionality reduction. Specifically, principal component analysis is investigated in order to test the impact of using less noisy data, but also to identify a limited set of "top" variables.

The efficacy of using the top variables only in the principal component analysis is tested by visually comparing labeled bi-plots of the 16 observations in order to observe the separability of the data. It is shown that using

the top 100 variables (gene expression levels in this case) does not improve this separability, but does not significantly worsen it.

Finally, we move beyond visual inspection to test how well our data clusters into two groups - hopefully matching what we know about the "good" and "poor" labels of our observations. Both agglomerative and K-means clustering approaches are conducted on the full data, and the top 100 gene data. For each set of data, six different scaling methods are used as input. Again, we see similar performance for the top 100 gene data (as was the case with PCA and visual inspection), but we also observe performance differences of the clustering approaches and the types of scaling. The performance is measured by observing the best possible accuracy from a confusion matrix of the assigned clusters and the known labels (e.g. the higher of 40% and 60% would be 60%). The key observations are:

- Clustering techniques: K-means generally performs better, and never worse, than agglomerative clustering.
- Scaling techniques: No particular technique performed best overall, but versions of log transforms of the data were consistently good performers.
- Data completeness: Using the full data as input we were able to find relatively accurate clusters regardless of clustering approach. Using the top 100 genes performed similarly, but consistently well for all scaling techniques under K-means clustering. The PCA data performed well with K-means clustering.

1.2 PCA analysis

1.3 Principal component analysis

Principal component analysis is able to systematically find vectors within the data that have the highest variance (and thus explain the most variation), and are orthogonal to other principal component vectors. The number of principal components is the lower of the the number of observations, or the number of variables. For this exercise the following method was used to find principal components:

- "prcomp" package used from "stats" library in R
- Only the centered and scaled data was used (none of the other scaling methods), because it's critical that each variable have a mean of 0 (James et al. 2013). The centered and scaled log transform data may be interesting to observe in future investigations.

There does appear to be some grouping between the two kinds of observations ("good" and "poor") when reviewing the first two components:

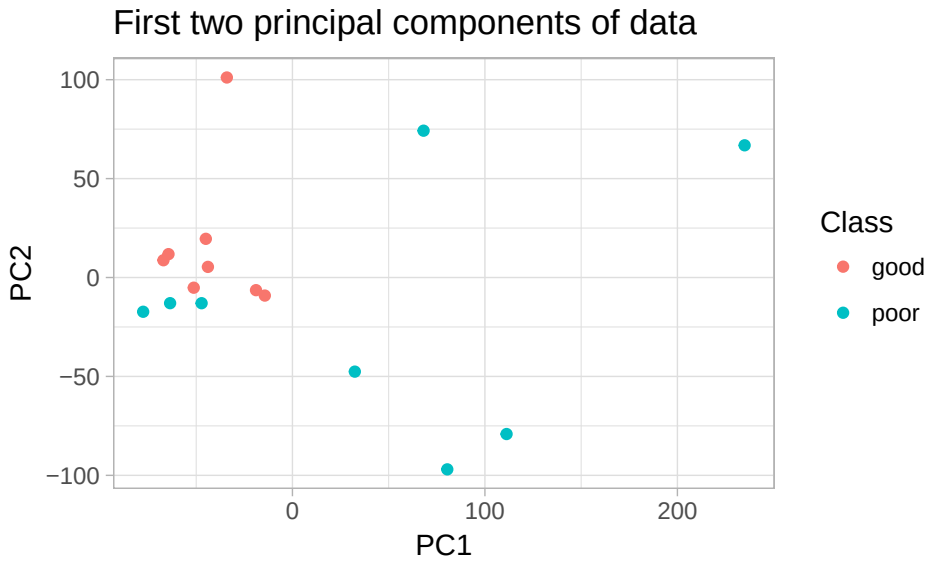

Figure 1: First two principal components trained from full dataset

Approximately 50% of the variance is captured in these first two components:

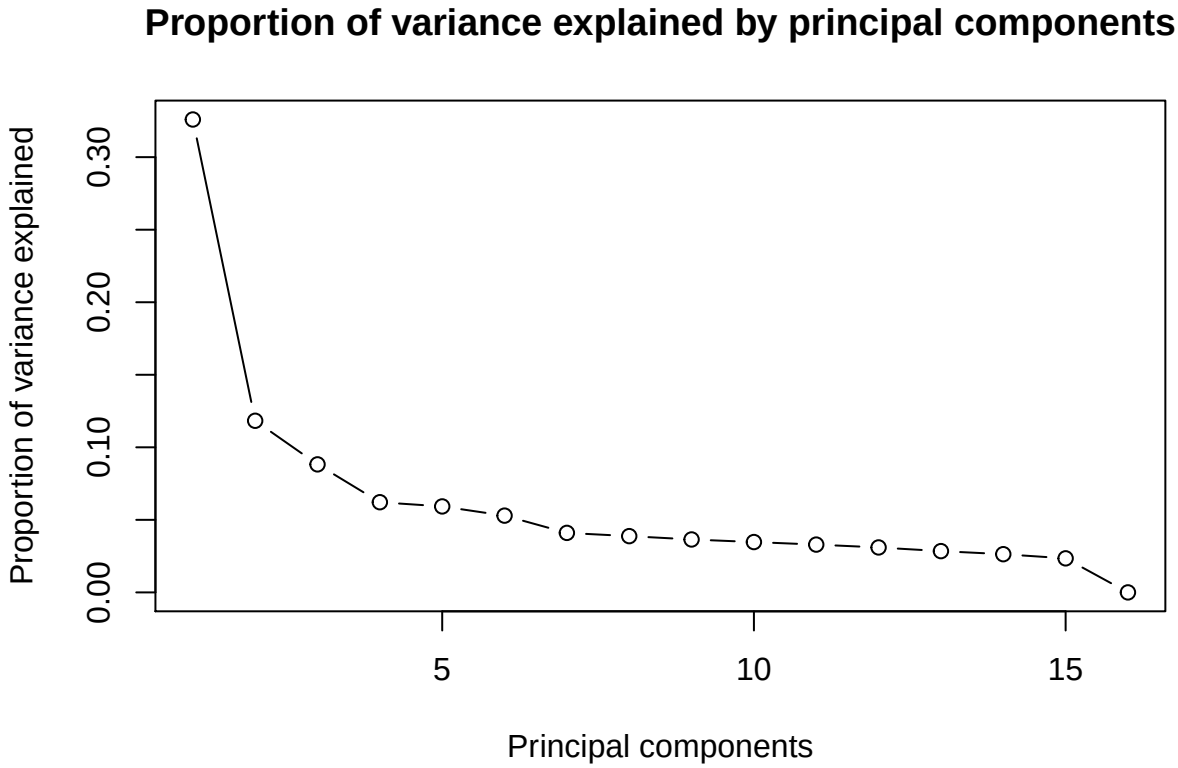

Figure 2: Proportion of variance explained by principal components

1.4 Using PCA results to identify the top 100 genes

The method used to find the top 100 genes from the PCA output is as follows:

1. Retrieve the loadings (rotation matrix) and square all values
2. Discard all PC's and keep the first only
3. Sort these squared values in descending order and identify the top 100

Because the most influential variables will have the highest absolute loadings on the principal components, the squared loadings will represent the most influential variables. Because the first principal component contains the most information, the variables that have the highest squared loadings therefore represent the variables that contribute the most to principal component with the most information and are subsequently considered the most important.

The top five genes are: X204365_s_at, X206766_at, X221870_at, X204542_at, X215356_at

These top 100 genes result in two distinct groups of observations (plus one far outlier), one of which is entirely made up of "poor" observations:

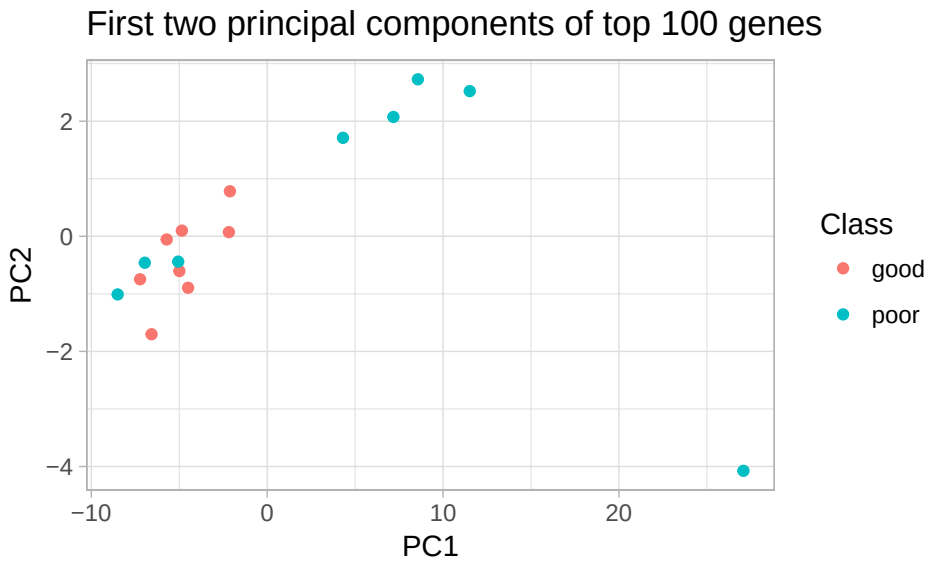

Figure 3: First two principal components trained from top 100 genes

When considering the proportion of variance explained by each principal component, it is not surprising that almost 90% of the variance is explained by PC1 - this is because we chose the variables that had the highest loadings for PC1. Very little information is contained in the other components.

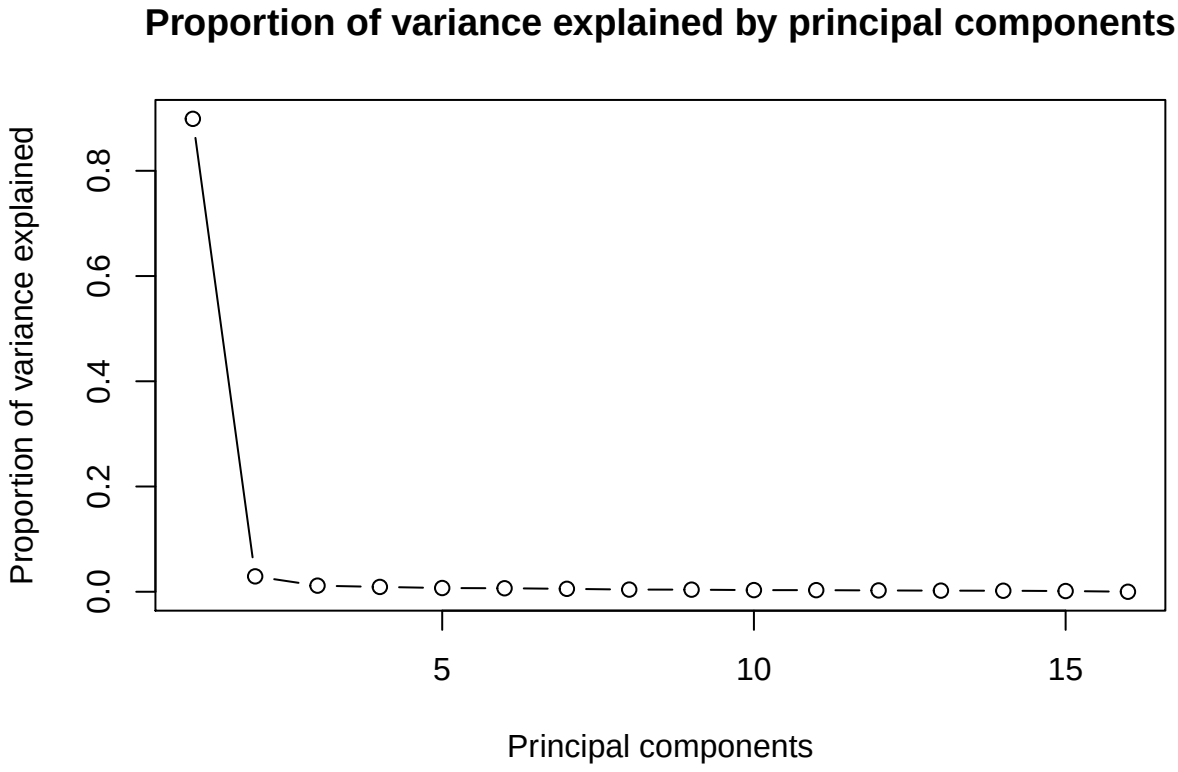

Figure 4: Proportion of variance explained by principal components

1.5 Clustering analysis

Next, clustering analysis is performed on the data. Two different kinds of clustering are used:

1. Agglomerative: This is a bottom-up approach that builds up clusters from 16 observations to a single cluster contained all observations. The "tree" is then cut to ensure two clusters. Only the "complete" method was used for this project after some initial testing indicated that it produced better results with more coherent trees.
2. K-means: Two clusters are specified and an iterative approach begins to adjust cluster centers and cluster allocations until the algorithm stabilizes.

Three different inputs were fed to the clustering algorithms:

1. Each of the six scaled full data-sets
2. Each of the six scaled top 100 gene data-sets
3. Principal components from full data-set

1.5.1 A note on scaling approaches

As mentioned, there were six scaling techniques used. They are as follows:

1. "Range 0-1 per column": The minimum and maximum for each column (variable) is used to transform each column's data to a number between 0 and 1.
2. "Range 0-1 for all data": The minimum and maximum for the whole data-set is used to transform all the data to a number between 0 and 1.
3. "Centered and scaled per column": This is the typical approach where for each column the mean is subtracted in order to center at 0, and then each column's data is divided by the column's standard deviation.
4. "Log transform": The natural log of each data point is used.
5. "Log transform and Range 0-1 per column": The "Range 0-1 per column" method is applied to the "Log transform" data.
6. "Log transform and centered and scaled per columns" The "Centered and scaled per column" method is applied to the "Log transform" data.

1.5.2 Hierarchical / agglomerative clustering full dataset

Most scaled data-sets result in quite a clear smaller branch of "poor" observations, with a larger mixed branch:

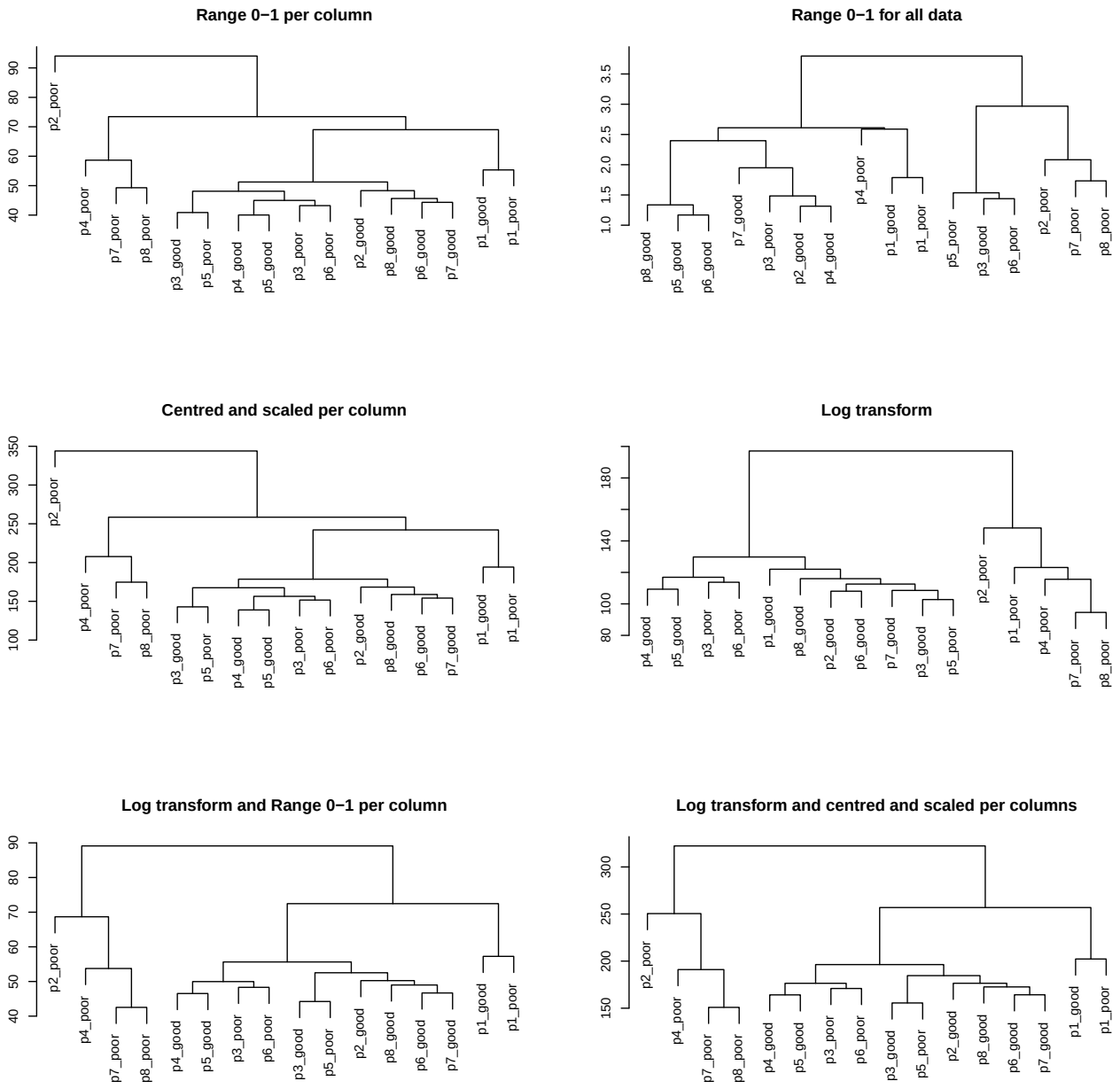

Figure 5: Dendrograms of six scaling methods

These dendrograms can be summarized neatly into confusion matrices. Because this was an unsupervised task, there is no definitive true positive or true negative region - hence we consider the diagonal with the most observations. Only the "log transform" data achieved the maximum classification accuracy of 13/16 (~81%):

Table 1: Confusion matrices per scaling method

```
## Range 0-1 per column:
## Assigned cluster good poor
##                1    8    7
##                2    0    1
##
## Range 0-1 for all data:
## Assigned cluster good poor
##                1    7    3
```

```
##                2    1    5
##
## Centred and scaled per column:
## Assigned cluster good poor
##                1    8    7
##                2    0    1
##
## Log transform:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform and Range 0-1 per column:
## Assigned cluster good poor
##                1    8    4
##                2    0    4
##
## Log transform and centred and scaled per columns:
## Assigned cluster good poor
##                1    8    4
##                2    0    4
```

1.5.3 K-means clustering full dataset

The K-means clustering approach was more consistent, achieving 13/16 classification accuracy for all scaling methods except for the "Range 0-1 for all data":

Table 2: Confusion matrices per scaling method

```
## Range 0-1 per column:
## Assigned cluster good poor
##                1    0    5
##                2    8    3
##
## Range 0-1 for all data:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Centred and scaled per column:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform and Range 0-1 per column:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
```

```
## Log transform and centred and scaled per columns:
## Assigned cluster good poor
##                1    0    5
##                2    8    3
```

1.5.4 Hierarchical / agglomerative clustering top 100 genes

Using the top 100 genes only, we see similar looking results to that of using the full data-set - a small branch of "poor" observations, and a large branch of mixed observations:

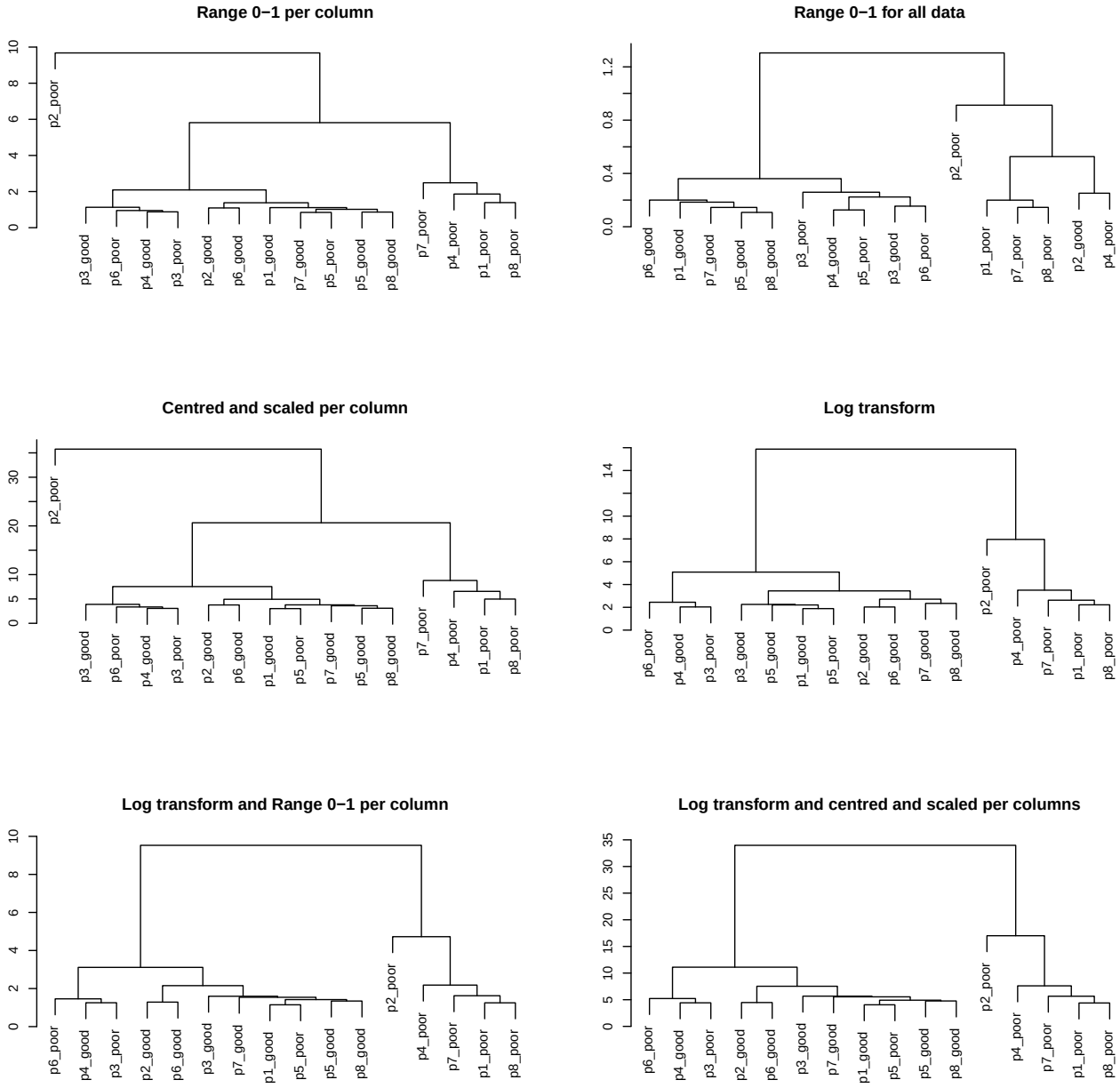

Figure 6: Dendrograms of six scaling methods

Reviewing the classification accuracy, it appears that a log transform of the data was very helpful as all scaling methods employing it performed at a 13/16 accuracy:

Table 3: Confusion matrices per scaling method

```
## Range 0-1 per column:
## Assigned cluster good poor
##                1    8    7
##                2    0    1
##
## Range 0-1 for all data:
## Assigned cluster good poor
##                1    7    3
##                2    1    5
##
## Centred and scaled per column:
## Assigned cluster good poor
##                1    8    7
##                2    0    1
##
## Log transform:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform and Range 0-1 per column:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform and centred and scaled per columns:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
```

1.5.5 K-means clustering top 100 genses

The K-means clustering on the top 100 data achieved 13/16 classification accuracy regardless of which scaling method was used:

Table 4: Confusion matrices per scaling method

```
## Range 0-1 per column:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Range 0-1 for all data:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Centred and scaled per column:
## Assigned cluster good poor
##                1    0    5
##                2    8    3
##
## Log transform:
```

```
## Assigned cluster good poor
##                1    0    5
##                2    8    3
##
## Log transform and Range 0-1 per column:
## Assigned cluster good poor
##                1    8    3
##                2    0    5
##
## Log transform and centred and scaled per columns:
## Assigned cluster good poor
##                1    0    5
##                2    8    3
```

1.5.6 Hierarchical / agglomerative clustering with principal component data

Using the principal component data results in a dendrogram with the "poor" outlier branch, and then a smaller three observation "poor" branch within the second branch. This is not as accurate as many instances of the original data without PCA:

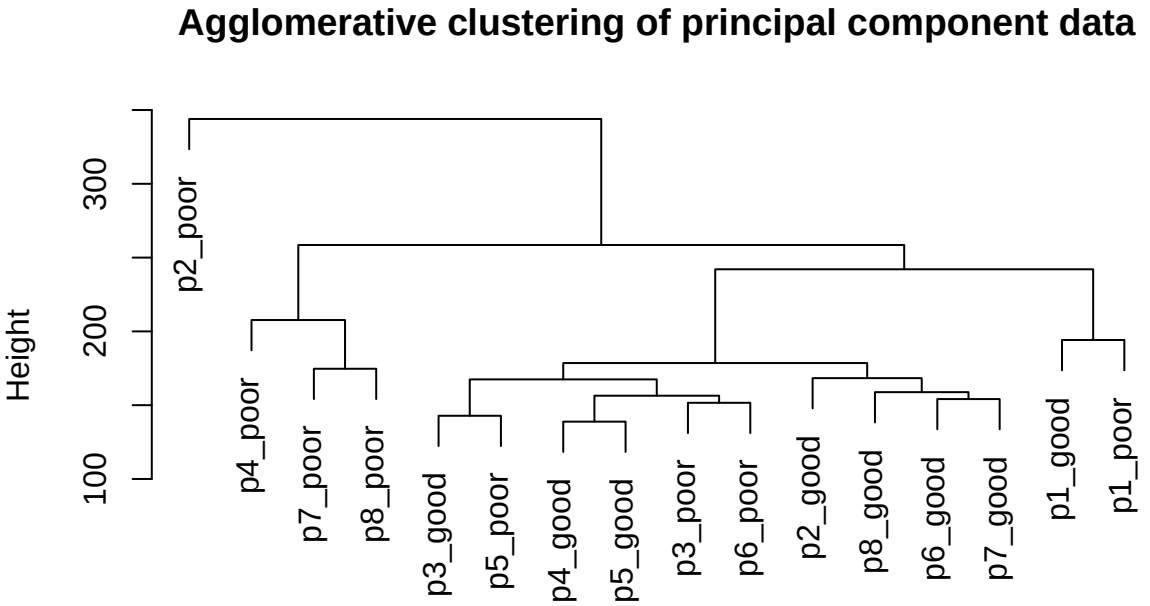

Figure 7: Dendrogram of principal component data

The result is an accuracy of 9/13, which is very likely to occur by random chance:

Table 5: Confusion matrix of principal component data

```
##
## Assigned cluster good poor
##                1    8    7
##                2    0    1
```

1.5.7 K-means clustering with principal component data

Using the K-means approach the PCA data is able to produce an accuracy of 13/16. In general, K-means has more consistently been accurate at the 13/16 level, so it does not seem that PCA was that helpful.

Table 6: Confusion matrix of principal component data

```
##
## Assigned cluster good poor
##                1    0    5
##                2    8    3
```

1.6 Conclusion

While no combination of selected input, scaling, or clustering was able to perform better than 13/16 accuracy, the results are interesting in that there are many combinations able to classify 5/8 "poor" observations into their own category. This could mean that gene expression levels may be analysed to help identify "poor" Leukemia prognosis with little chance of a false positive, but a fairly high chance of false negative - i.e. high precision, but a high false negative rate.

2 Project 2: New vehicle dataset

2.1 Introduction

This section of the assignment makes use of a data-set of motor vehicles - specifically 2016 models available in South Africa. The goal of the analysis is to try and identify groups of vehicles which exhibit similar characteristics, which may be a useful way to potentially direct someone who was trying to navigate which set of vehicles to choose from.

The data is originally from Transunion, and has been cleaned to ensure that:

- Only 2016 vehicles are considered
- There are no duplicates within the data
- Only interesting variables are considered (e.g. Cooling system is not included, but price is). 34 variables are included in total.

It is expected that certain groups of vehicles will cluster: e.g. expensive sports cars in one group; "bakkies" in one group; small "cheap" cars in another.

2.2 Principal component analysis of data

In order to better understand the data, a principal component analysis on scaled data is performed. From the bi-plot it can be seen that the observations are fairly evenly distributed along the first two components indicating that there could be some distinct clusters within the data. The proportion of variance explained shows that most of the variance is explained by the first five components, although the remaining components are not negligible.

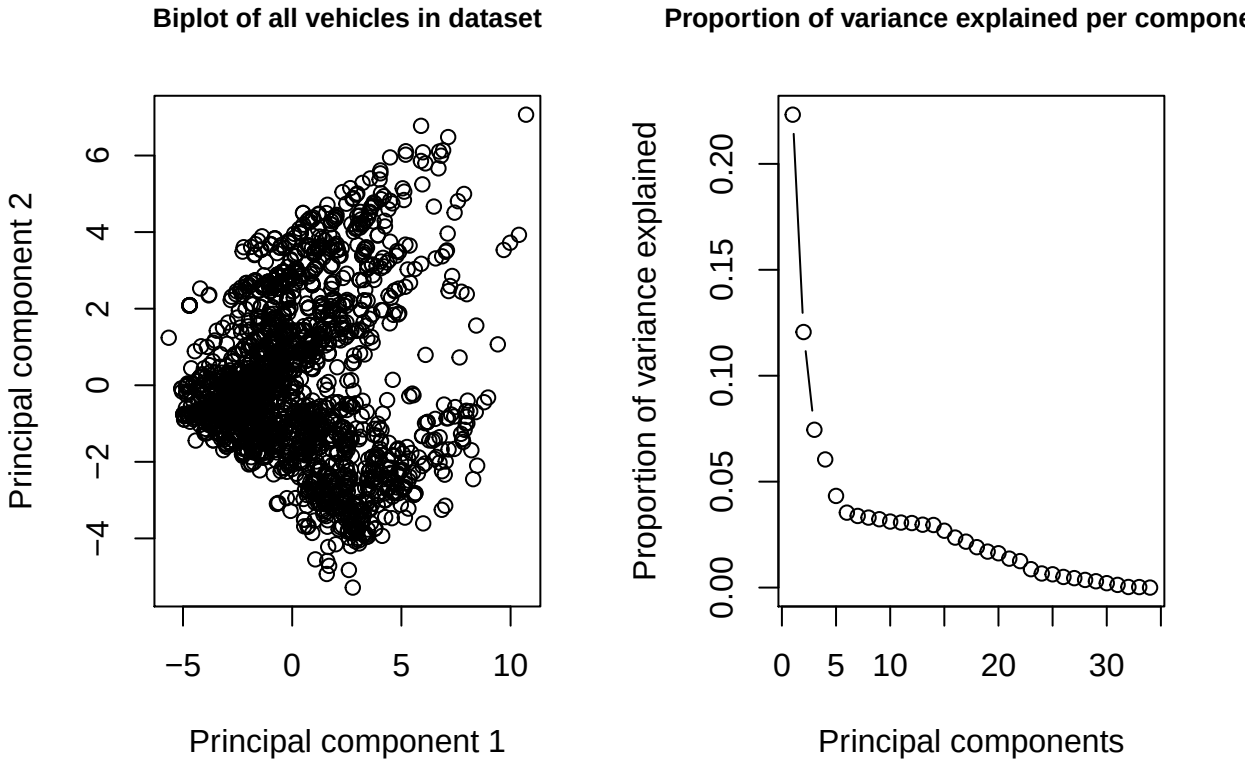

Figure 8: Biplot and variance explained

2.3 Clustering using K-means

In order to better observe the representation of the data in a 2-dimensional space, the data is clustered into 4 groups using K-means performed on scaled data (centered and adjusted by standard deviation). Four groups was decided on after some initial experimentation, and standard scaling was chosen for simplicity.

2.4 Dimensionality reduction

2.4.1 t-SNE dimensionality reduction

The first dimensionality reduction approach was the fairly modern t-SNE (t-distributed stochastic neighbor embedding), which is an approach developed in 2008 by Geoffrey Hinton and Laurens van der Maaten (Van Der Maaten and Hinton 2008). It is similar to other multi-dimensional scaling approaches, except that it makes use of a probability distribution in order to select points. The major downside of t-SNE is that it can sometimes display patterns resulting from random noise - often multiple plots with different parameters should be analysed before making any expensive decisions based on it.

The four clusters are fairly distinct in this representation, but the t-SNE clumps indicate that further analysis could yield other clusters based off the t-SNE data itself:

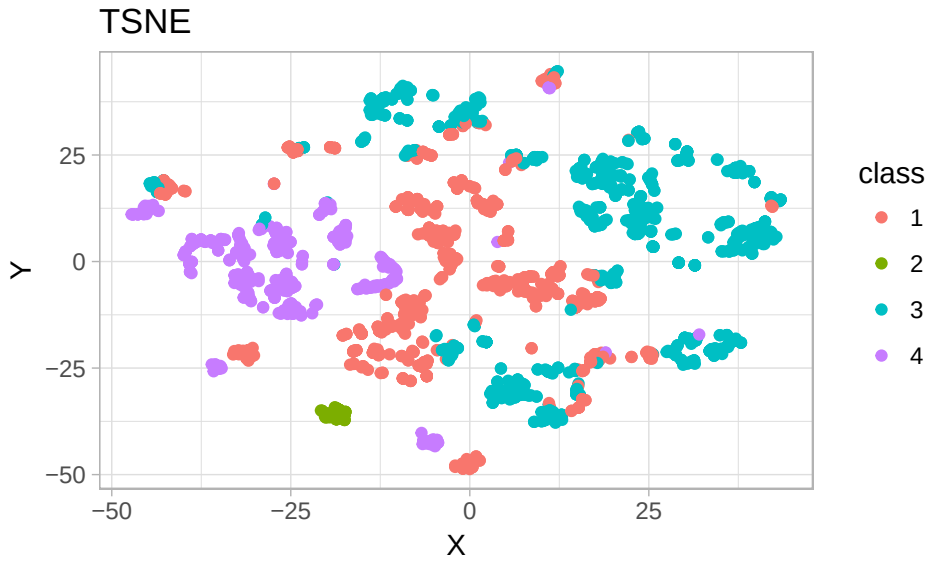

Figure 9: 2-dimensional t-SNE of data

2.4.2 MDS: CMDSCALE (classical scaling)

Classical scaling is a metric MDS approach (approximate inter-sample dissimilarities as closely as possible). The shape of the data here is quite similar to the PCA bi-plot. The clusters are quite clearly separated, with no strong outliers:

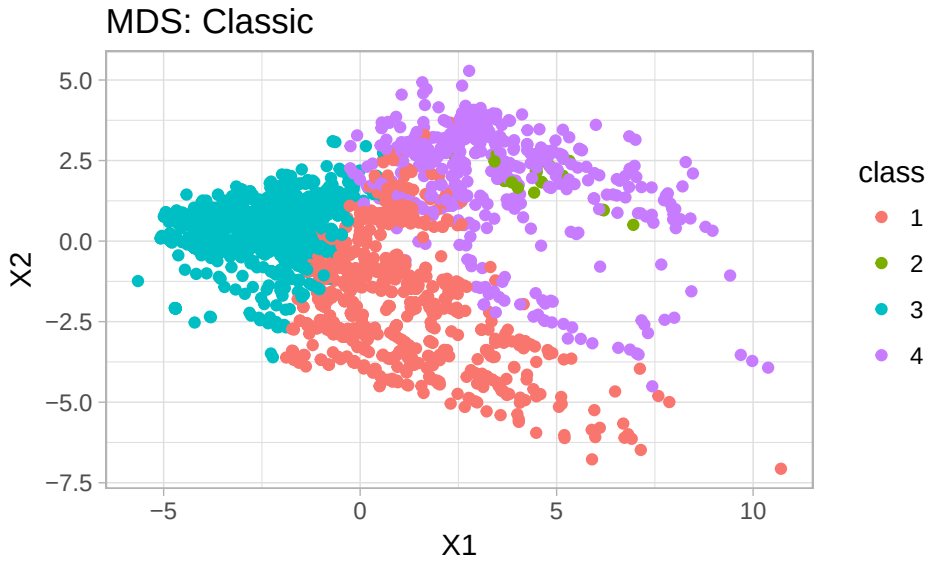

Figure 10: Classical metric MDS

2.4.3 MDS SMACOF Metric

The metric SMACOF approach is similar to the classic approach, but uses majorization to minimize the cost function. It is known to be an inefficient algorithm, and can the results show this: many outliers and a big clump in the center:

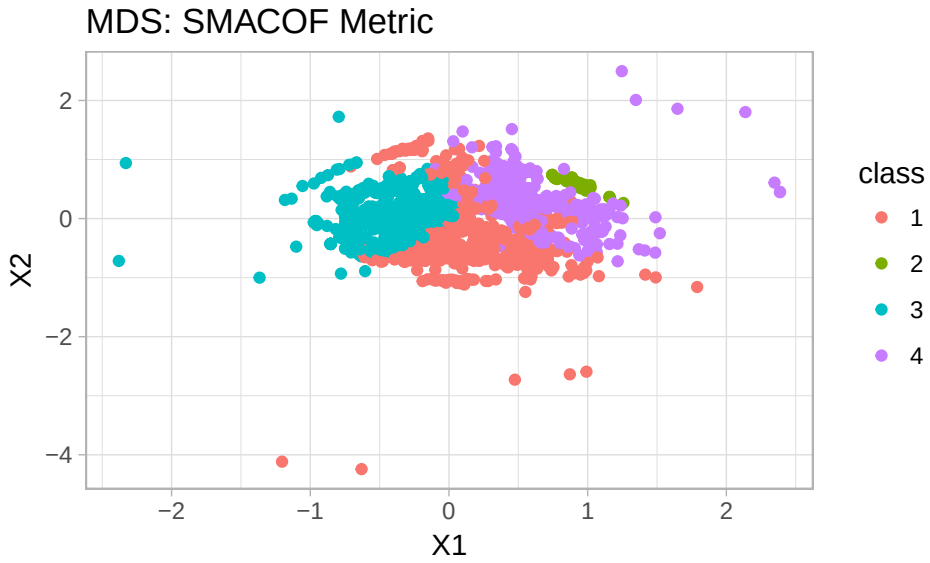

Figure 11: SMACOF Metric MDS

2.4.4 MDS SMACOF Non-metric

The non-metric (ordinal approach) version of SMACOF performs even worse on the data - the majority of observations are squeezed into the top right corner with a few outliers in the bottom left:

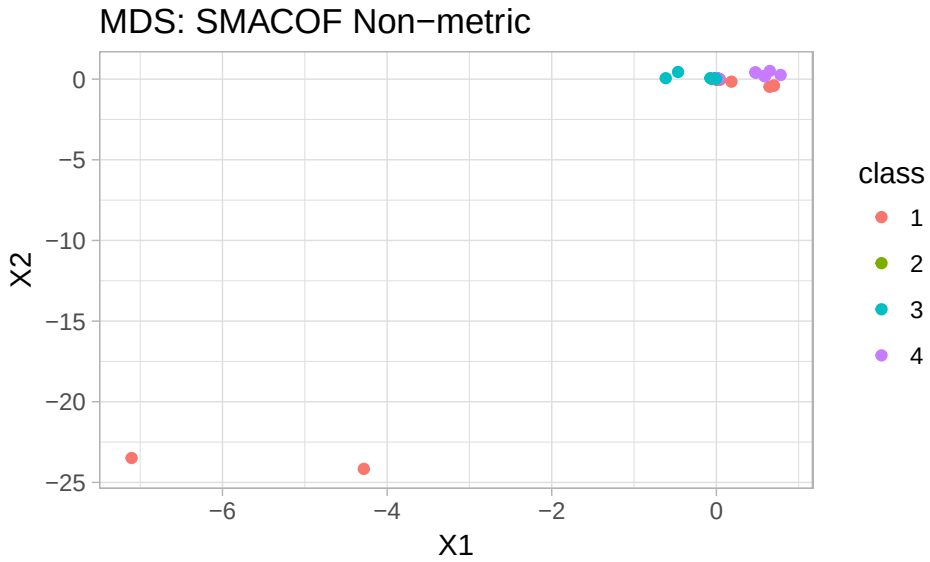

Figure 12: SMACOF Non-metric MDS

2.4.5 MDS: Kruskals non-metric

Kruskals is a non-metric approach that tries to minimize the discrepancy between the rank order of the full dimension distance, and the 2-dimension distance. While the clusters look somewhat separable, it is not as clean as the classical metric approach:

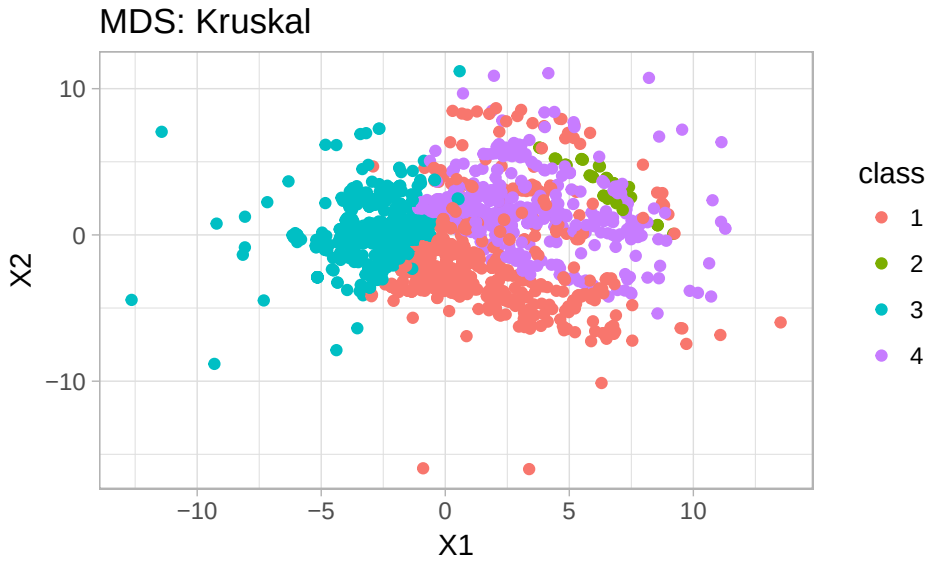

Figure 13: Kruskals Non-metric MDS

2.4.6 MDS: Sammon non-metric

The Sammon approach is very similar to Kruskals except that the error is adjusted further by the object distance in the original space. This seems to help the algorithm, as cleaner cluster separation is observed than Kruskals (although many outliers remain):

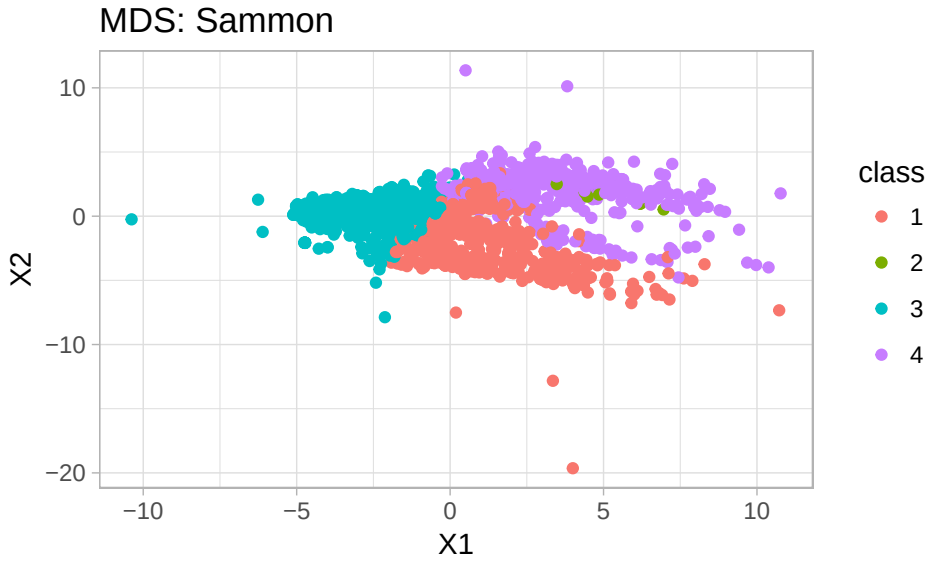

Figure 14: Sammon non-metric

2.5 Self-organising maps

Self organizing maps are an unsupervised neural network that allocates observations to neurons in a grid, while also adjusting grid parameters. It is a helpful technique that allows us to capture the characteristics of each neuron in full dimensionality while still being able to represent data in a 2-dimensional map. Neurons from a self-organizing map (or "Kohonen map") can be clustered themselves, and so the K-means clusters are not considered in this section.

2.5.1 9 nodes

The first map considered is a 3x3 hexagonal grid. The data is presented 1000 times, and the learning rate moves from 0.05 to 0.01. The following is observed:

1. The training process stabilized after about 400 iterations
2. There are two nodes with most of the observations (makes sense seeing that most vehicles are very similar, with a few specialty vehicles)
3. Looking at just two attributes (Kilowatts and price), these seem to correspond to the same vehicles and they are in the top right node

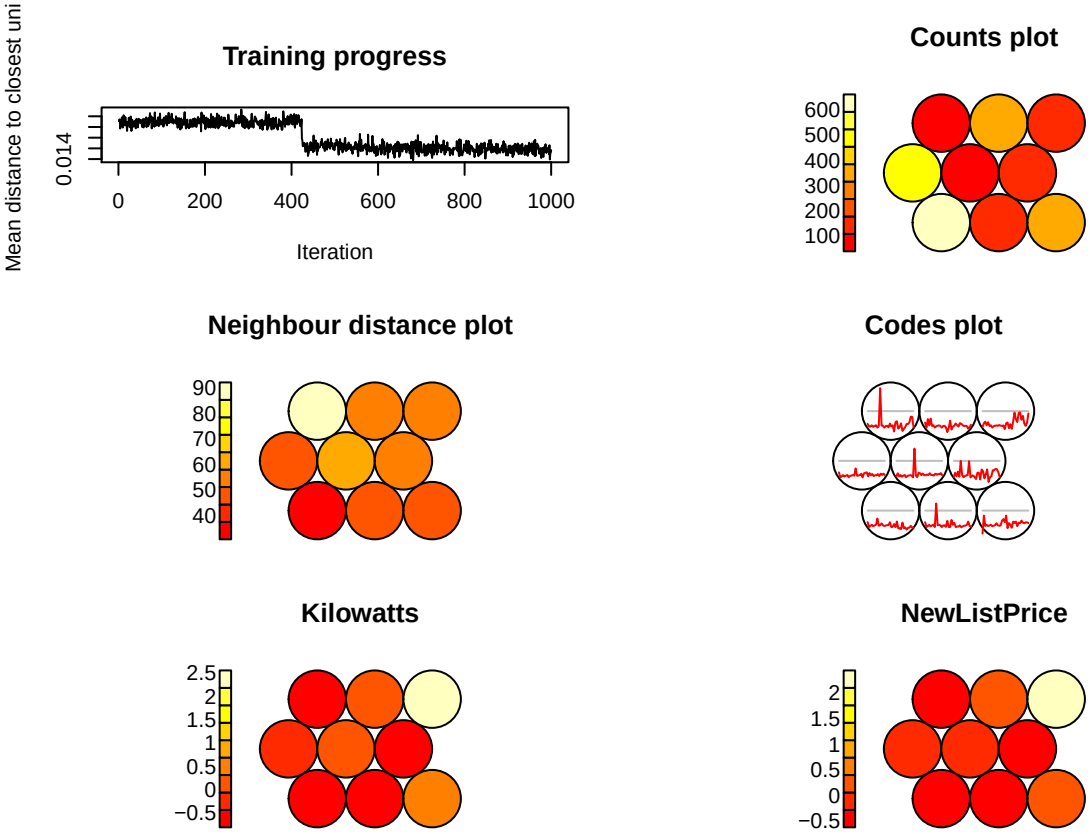

Figure 15: Key charts for 9 SOMS

Looking at the observations within each of the nodes (only a limited number of the 2000+ total observations are plotted), it does look like similar types of vehicles have been grouped:

- In the grey node in the top left there are a lot of delivery van type vehicles are grouped
- In the yellow node there are mostly "bakkies"
- The purples nodes seem to contain the most expensive premium vehicles, while the top purple node has the most powerful of these (5.0l vehicles)

More clusters could have been chosen, but four was selected in order to match the MDS section with K-means.

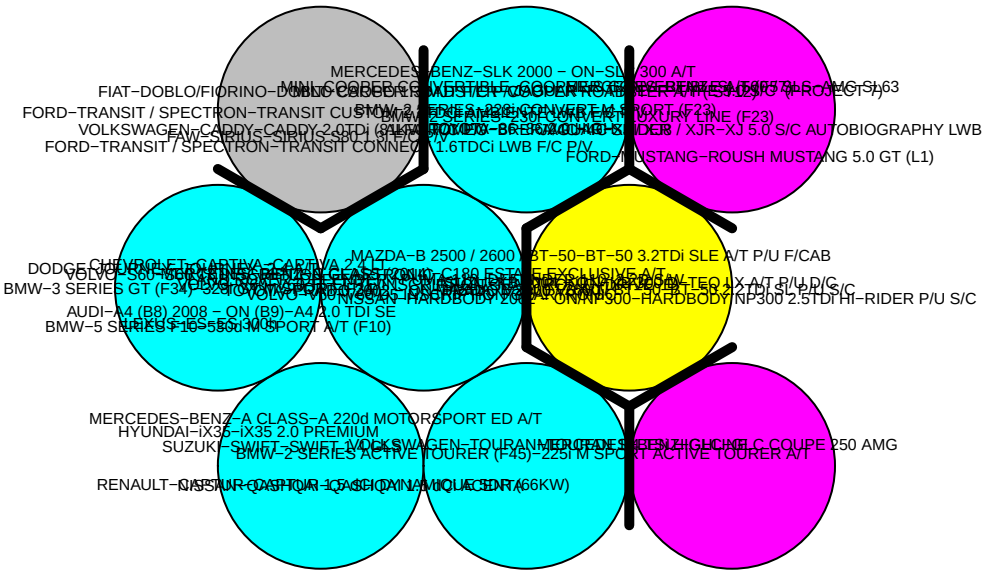

Figure 16: Mapping and clusters for 9 SOMS

2.5.2 4 nodes

A 2x2 grid is used with the same parameters as the 3x3 SOM. The following is observed:

1. The training process does not seem to stabilize as well as 3x3 grid
2. Expensive powerful cars still move to the same nodes, but are split across two nodes instead of just one as in the 3x3 grid
3. There is a more even distribution of observations between the nodes

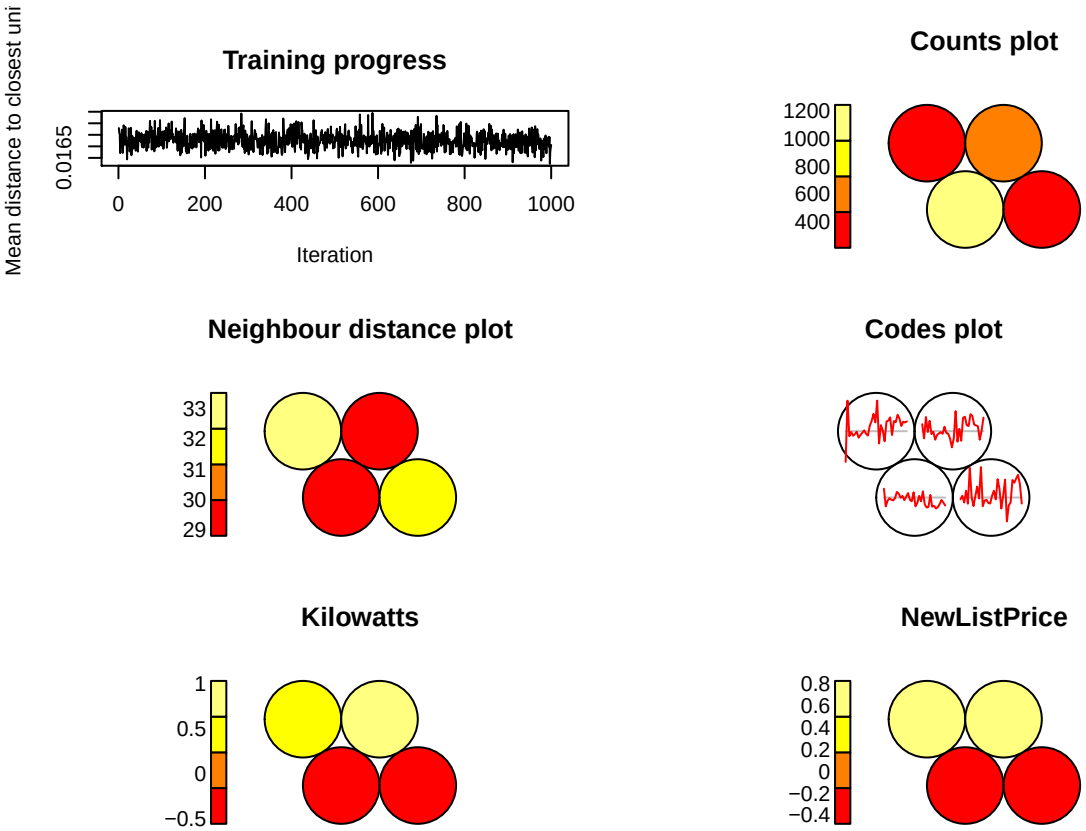

Figure 17: Key charts for 4 SOMS

Compared to the 3x3 map, there is a far less discernible pattern within the nodes. It is difficult to say in general what each node contains.

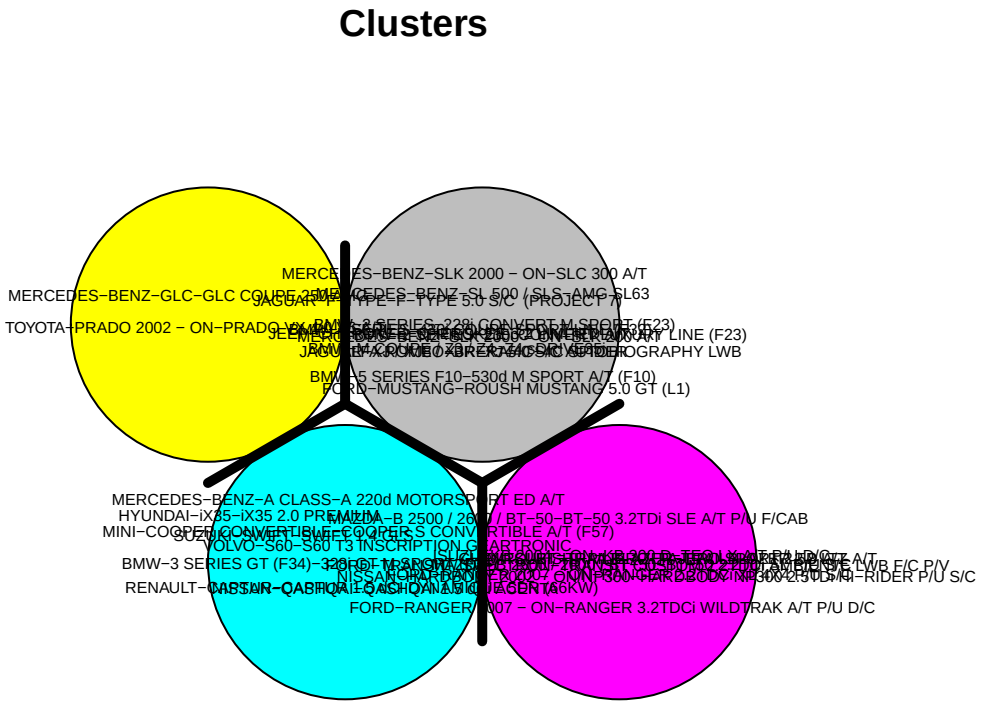

Figure 18: Mapping and clusters for 4 SOMS

2.5.3 Conclusion

According to the literature, the ideal map should contain about 5-10 observations per node (Lynn 2014). This would require an approximately 20x20 map which would require a more delicate plotting mechanism - potentially an interactive map - that would not work well in this format. However, even the 3x3 map was able to show some sensible groupings of vehicles indicating that it is a good technique for finding groups within complex product data.

2.6 Overall conclusion

The goal of project 2 was to be able to identify and visualize the vehicle data in such a way as to be able to sensible groups within the data. While it was not investigated further, the t-SNE approach showed potential for finding more than 4 clusters of data. However, the other MDS approaches were only able to show the 4 clusters and did not look as though they would be useful for identifying more.

The most successful approach tested was the larger SOM (3x3 grid) which was able to show sensible groups of vehicles. Further work should look at identifying clusters from the t-SNE output, and finding an optimal SOM grid size to further refine groups.

3 References

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

R. Wehrens and L.M.C. Buydens, Self- and Super-organising Maps in R: the kohonen package J. Stat. Softw., 21(5), 2007

Jesse H. Krijthe (2015). Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation, URL: https://github.com/jkrijthe/Rtsne

Jan de Leeuw, Patrick Mair (2009). Multidimensional Scaling Using Majorization: SMACOF in R. Journal of Statistical Software, 31(3), 1-30. URL http://www.jstatsoft.org/v31/i03/.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2016). cluster: Cluster Analysis Basics and Extensions. R package version 2.0.5.

Hadley Wickham (2017). tidyverse: Easily Install and Load 'Tidyverse' Packages. R package version 1.1.1. https://CRAN.R-project.org/package=tidyverse

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 103. Springer Texts in Statistics. New York, NY: Springer New York. doi:10.1007/978-1-4614-7138-7.

Lynn, Shane. 2014. "Self-Organising Maps for Customer Segmentation using R | Shane Lynn," 1–49. https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/ http://shanelynn.ie/index.php/self-organising-maps-for-customer-segmentation-using-r/.

Van Der Maaten, Laurens, and Geoffrey Hinton. 2008. "Visualizing Data using t-SNE." *Journal of Machine Learning Research* 9: 2579–2605. https://lvdmaaten.github.io/publications/papers/JMLR{_}2008.pdf.