Data Mining project - part II

Katarzyna Macioszek, Ada Majchrzak

February 5, 2024

1 Introduction

In the second part of the project we will use cluster analysis in connection with dimension reduction methods on *Spambase* dataset. Our goal is to asses clustering algorithms performance in differentiation between *spam* and *nonspam* types. For that we shall use *k-means*, PAM and AGNES clustering algorithms. We will also evaluate the quality of obtained results by checking separation measures of clusters built using original data and data after dimensionality reduction done with PCA algorithm. Additionally, we will also compare the results of classification performed on the former part of the project versus how the same methods will perform after applying dimensionality reduction.

2 Data preparation

To be consistent with analysis done in previous part of the project we again exclude columns that exhibit high correlation with each other. These eliminated features are *num415* and *num857*. Furthermore, we also use the same data transformations: z-score standardization and $\log(x + 0.1)$ transformation to make the explanatory variables have comparable values, as it is crucial for clustering and dimensionality reduction algorithms.

3 PCA

To be able to visualize the multidimensional data easily and reduce the noise in our data before applying the clustering algorithms, we start with performing the principal component analysis (PCA). As mentioned before, we use two data transformations, however to display an interesting behavior of the method, let us first present the outputs for non-transformed data. To visualize PCA results we will use scree plot (Figure 1) and contributions of variables to the first two principal components (Figure 2).

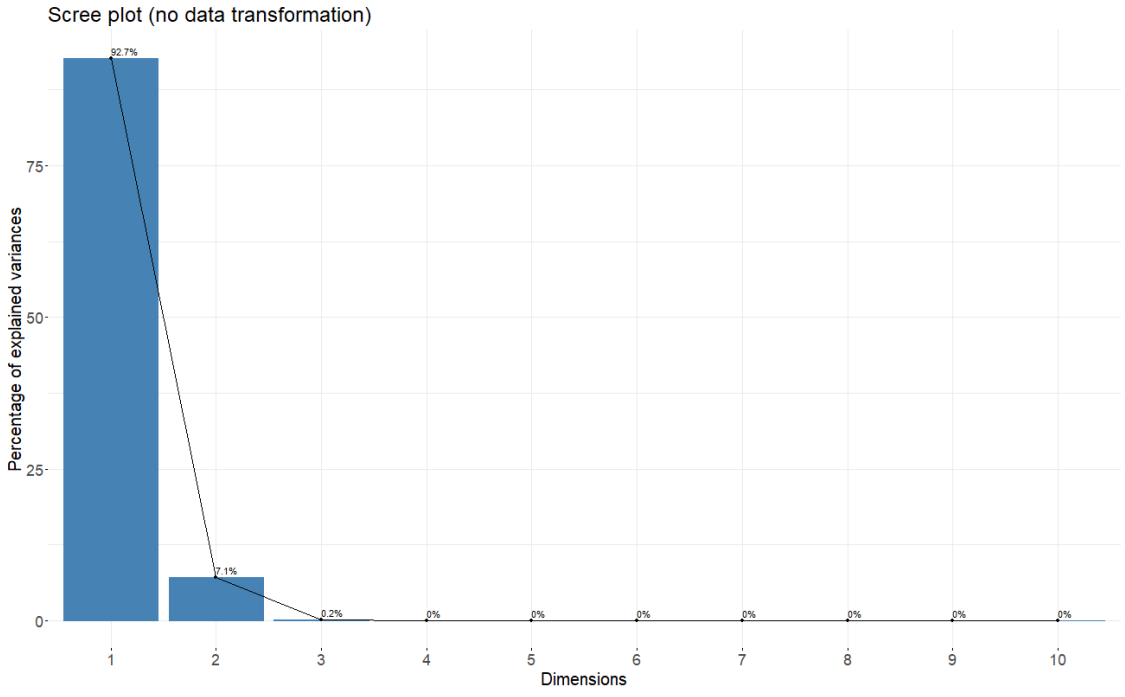

Figure 1: PCA results for original data (scree plot)

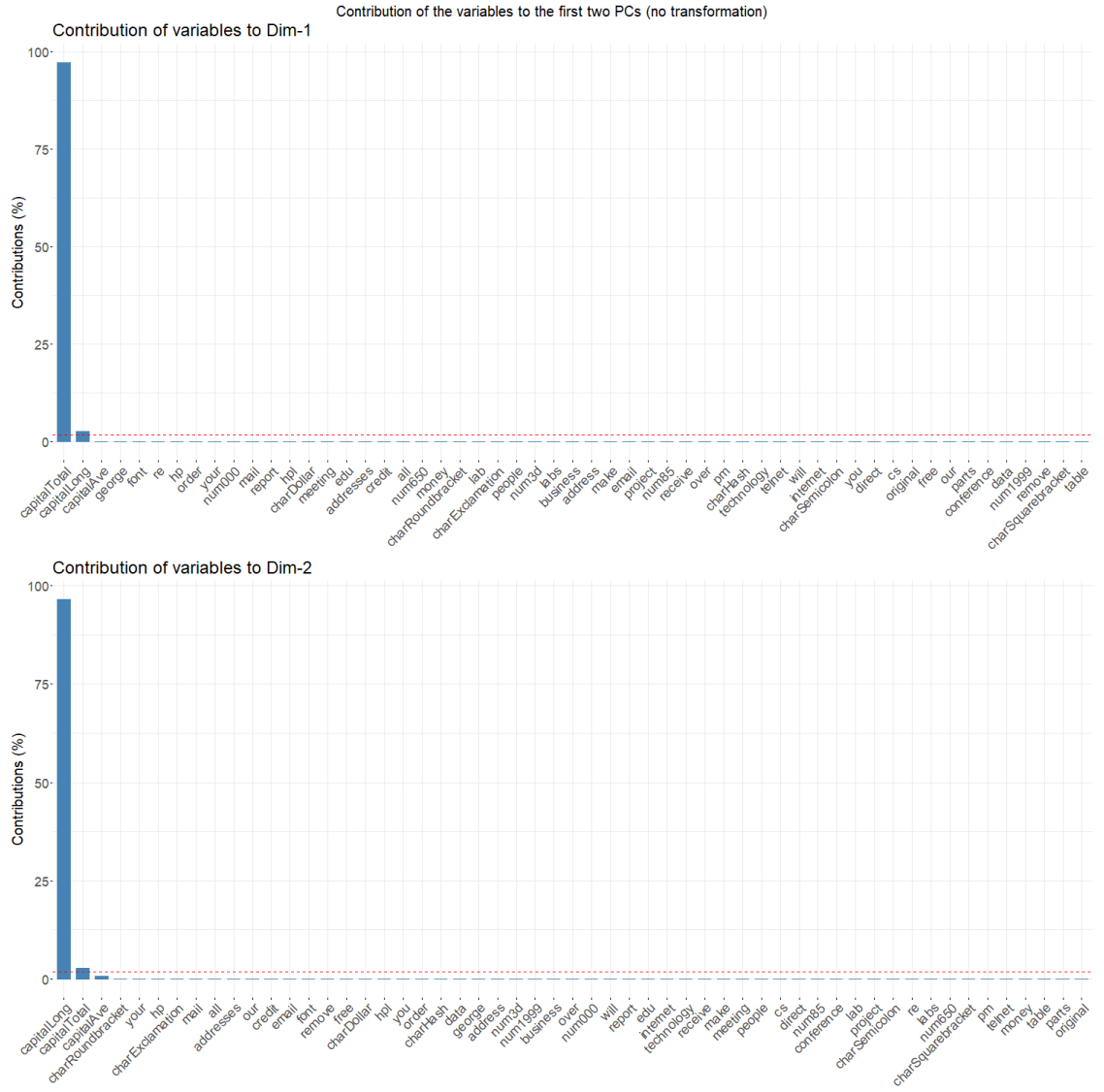

Figure 2: PCA results for original data (variable contribution plot)

As we expected, the algorithm gives poor results as only variables with highest spread are taken into consideration. In our data set most of the variables are frequencies with values in interval $[0, 100]$ and relatively low variance, and only capital letters characteristics have wider range of values. That causes the observed behavior of PCA method, which results with only 2 principal components, so in the following analyses we will use only transformed data.

Let us now check the principal components for data after z-score standardization. The screeplot is displayed on Figure 4, while the variable contribution plot can be found on Figure 4. As we anticipated, in this case we get more principal components created and also increased number of features is taken into account.

Results for log-transformed data can be seen on Figure 5 (scree plot) and Figure 6 (variable contribution plot). Here again we observe that contribution of many variables is higher than for original data. Also we see that in comparison to z-score transformed data we will probably be able to eliminate more principal components for the latter analysis.

To reduce dimensionality of our data using PCA, for both data transformations we selected a method based on feature variance. First, we compute mean standard deviation across all principal components. Second, we check which PCs display lower standard deviation than the computed mean. Finally, we reject those PCs falling below threshold – as a result, we obtain 26 principal components for z-score standardized data and 19 of them for log-transformed data. This seems like a really good reduction in dimensionality, considering that we started off with as much as 55 features!

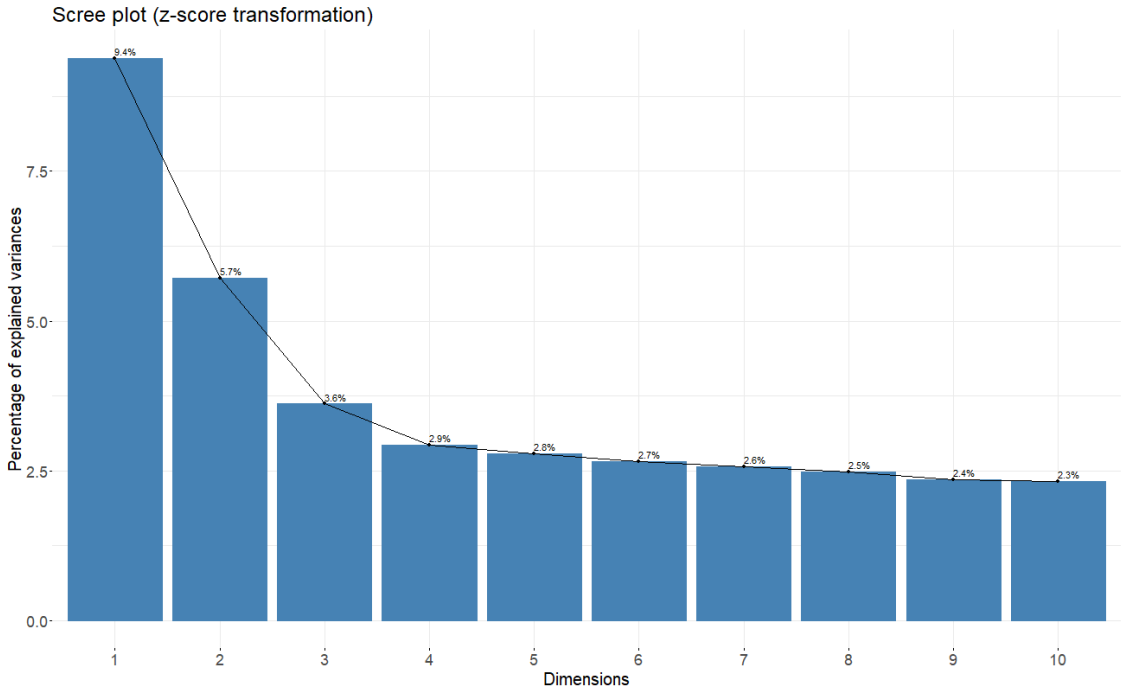

Figure 3: PCA results for standardized data (scree plot)

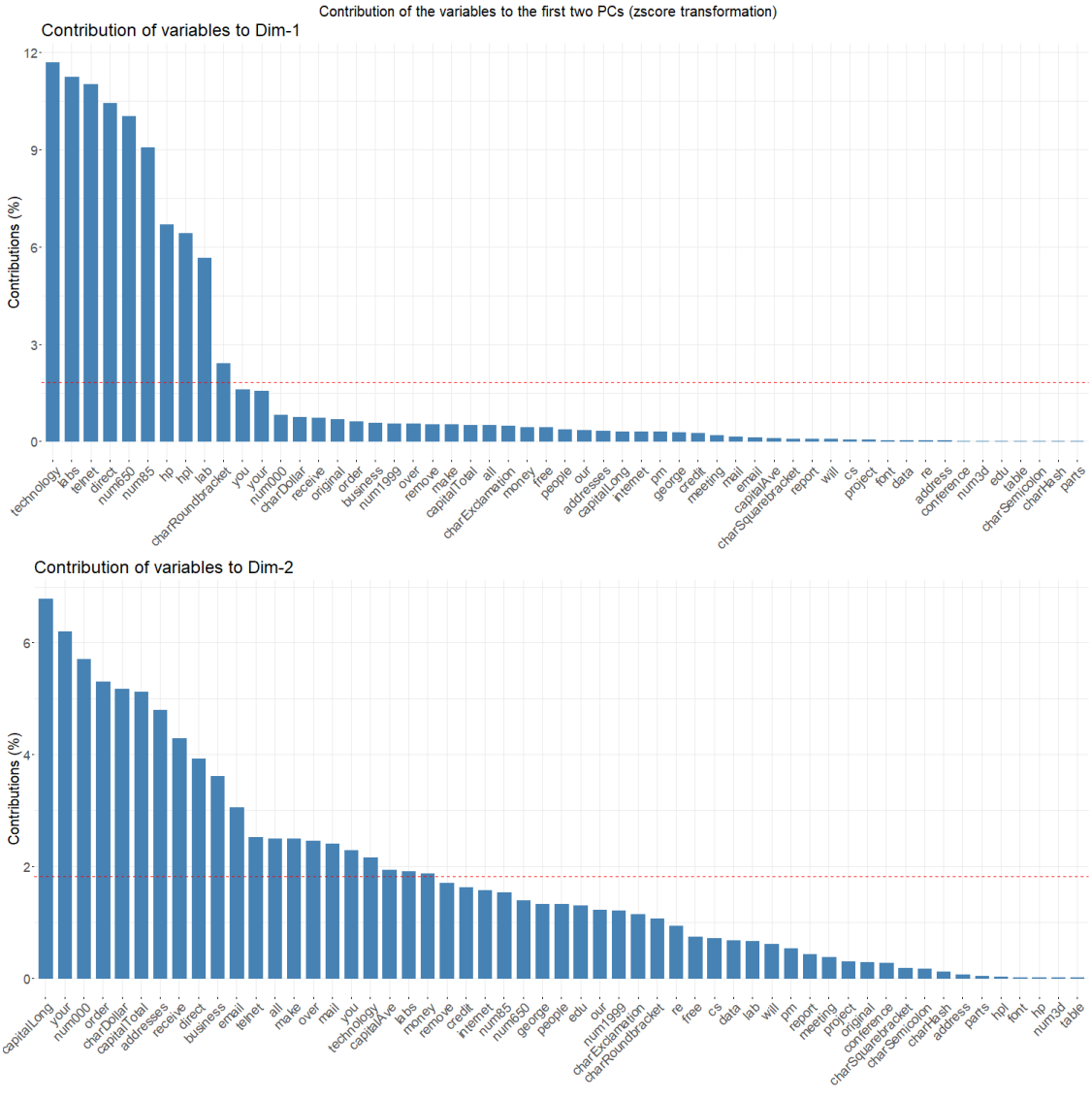

Figure 4: PCA results for standardized data (variable contribution plot)

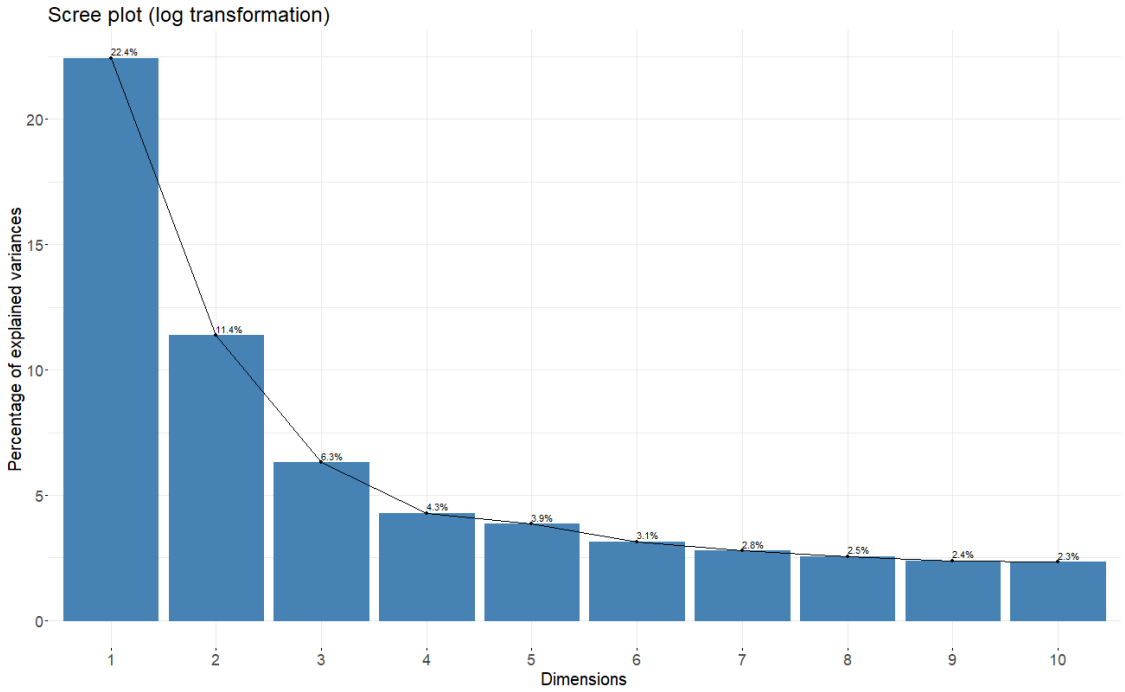

Figure 5: PCA results for log-transformed data (scree plot)

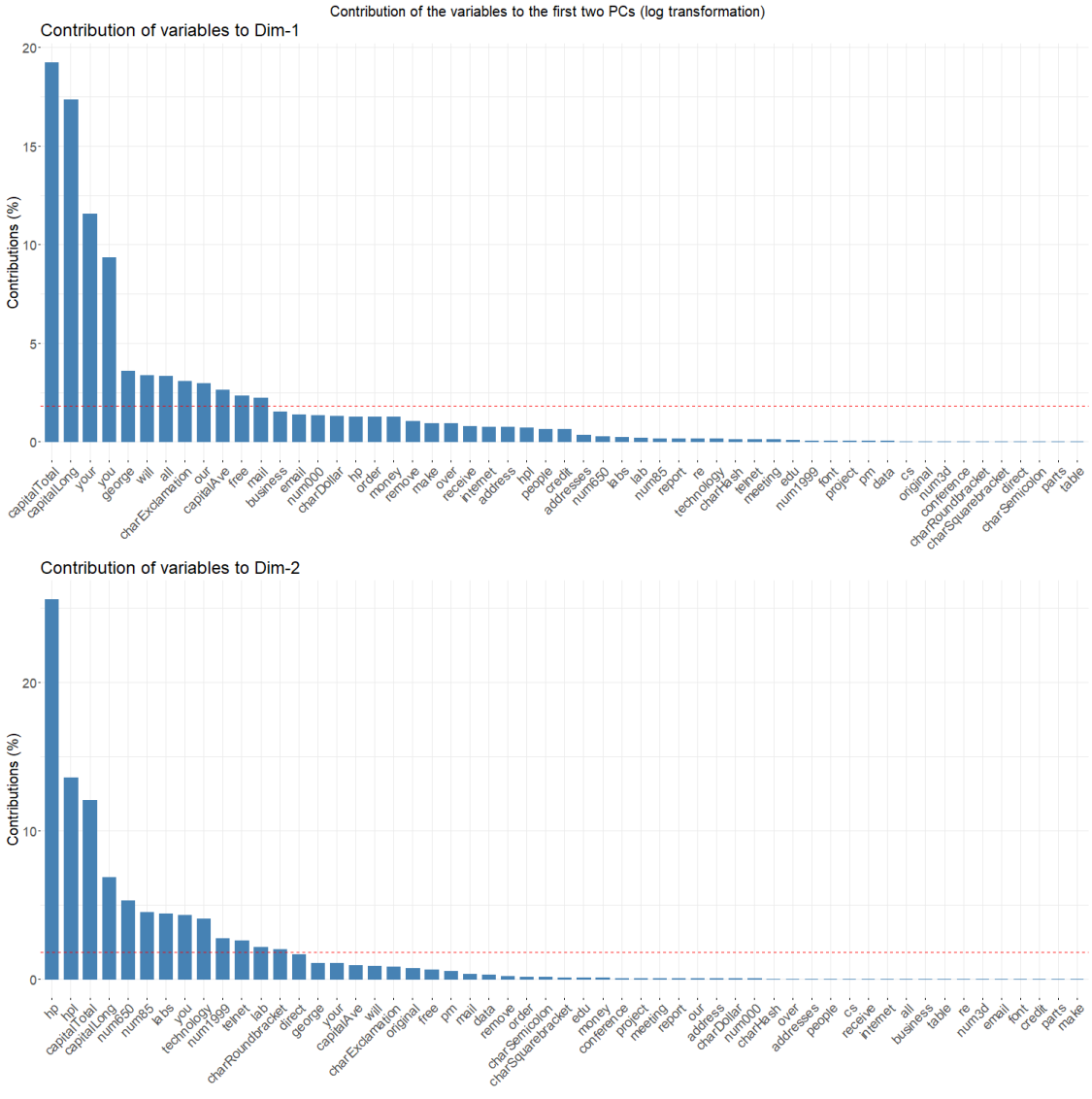

Figure 6: PCA results for log-transformed data (variable contribution plot)

4 Clustering

For cluster analysis we will use k-means, PAM and AGNES algorithms for transformed *Spambase* datasets and also for datasets built using principal components extracted while performing PCA. First thing we need to do before applying the algorithms is checking the optimal number of clusters k for each of them. In case of k-means and PAM, we will do it by calculating the silhouette index, and for AGNES we will select the number of clusters by analysing the dendrogram. Given the fact that the main goal of our analysis is to check how well the clustering algorithms will be able to differentiate between spam and non-spam e-mails, we expect the optimal number of clusters to be $k = 2$, but of course we cannot eliminate other possibilities, as there might be some hidden data structures within our dataset which should not be omitted.

Let us first take a look at silhouette index values for k-means without PCA, shown on Figure 7. We see that for the z-score standardized data we should extract 2 clusters, while for the log-transformed data the optimal number of clusters is 3. The same is observed for k-means with PCA (Figure 8). Situation changes slightly in case of PAM algorithm – here, the silhouette index method picked $k = 2$ for both data transformations without PCA (Figure 9), and for PCA $k = 3$ (Figure 10).

When it comes to AGNES, the choice of the optimal number of clusters is less obvious. Looking at the dendrogram, we need to pick such k that the increase in between-cluster dissimilarity from k clusters to $k+1$ clusters is significantly less than the increase in between-cluster dissimilarity from $k-1$ clusters to k clusters. From Figure 11 we see that for z-score standardized data without PCA $k = 2$ would be a good choice ($k = 3$ to include outliers as separate cluster), and the same with PCA (Figure 12). For log-transformed data, both without PCA (Figure 13) and with PCA (Figure 14), the optimal number of clusters is evaluating to 3. As expected, in case of the second data transformation we don't observe any outliers dectected by AGNES.

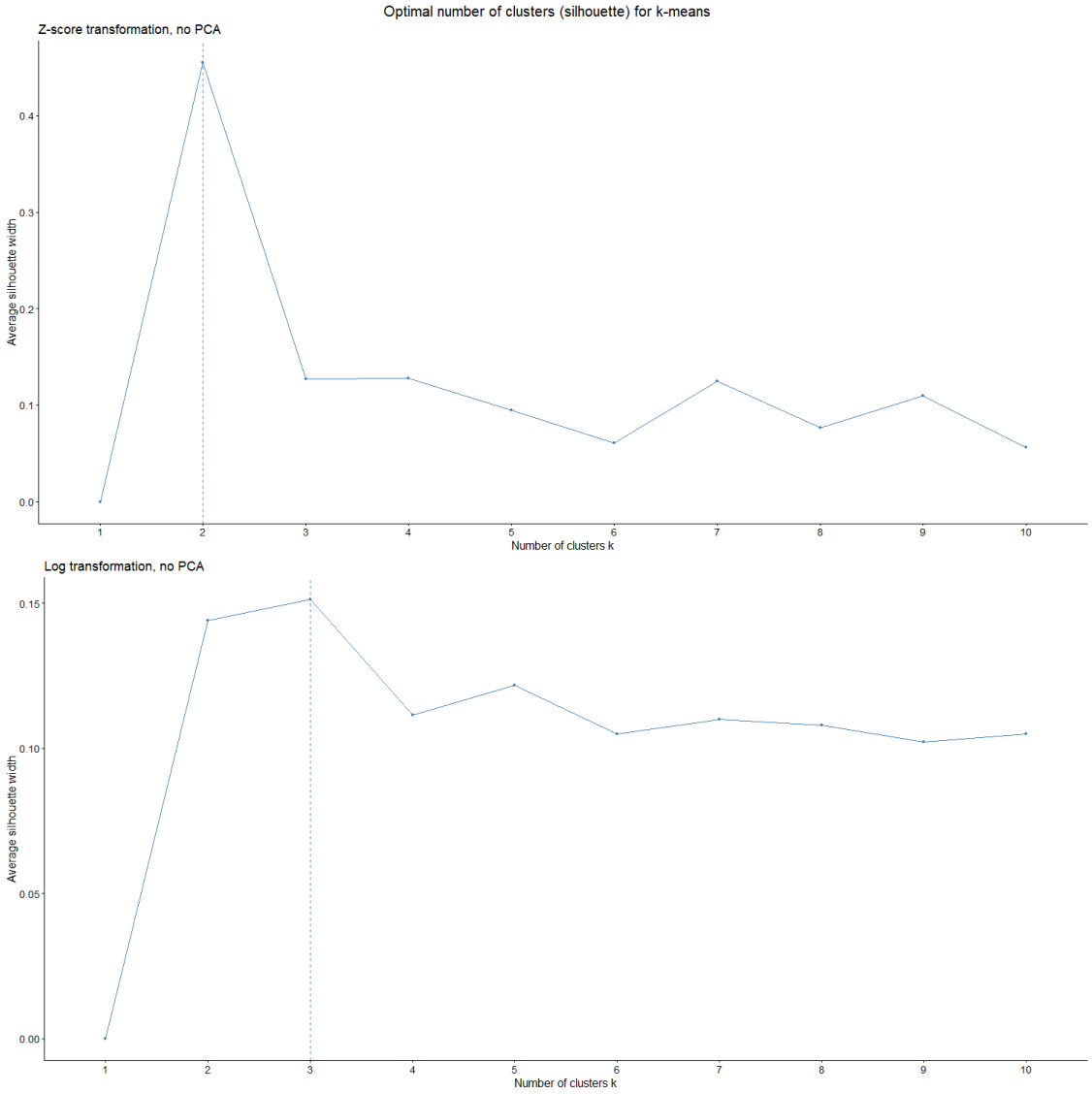

Figure 7: Number of clusters for k-means (no PCA)

Once we have selected the optimal number of clusters k for all algorithms, we can move on to clustering. In the following subsections, we will discuss the results obtained for k-means, PAM and AGNES.

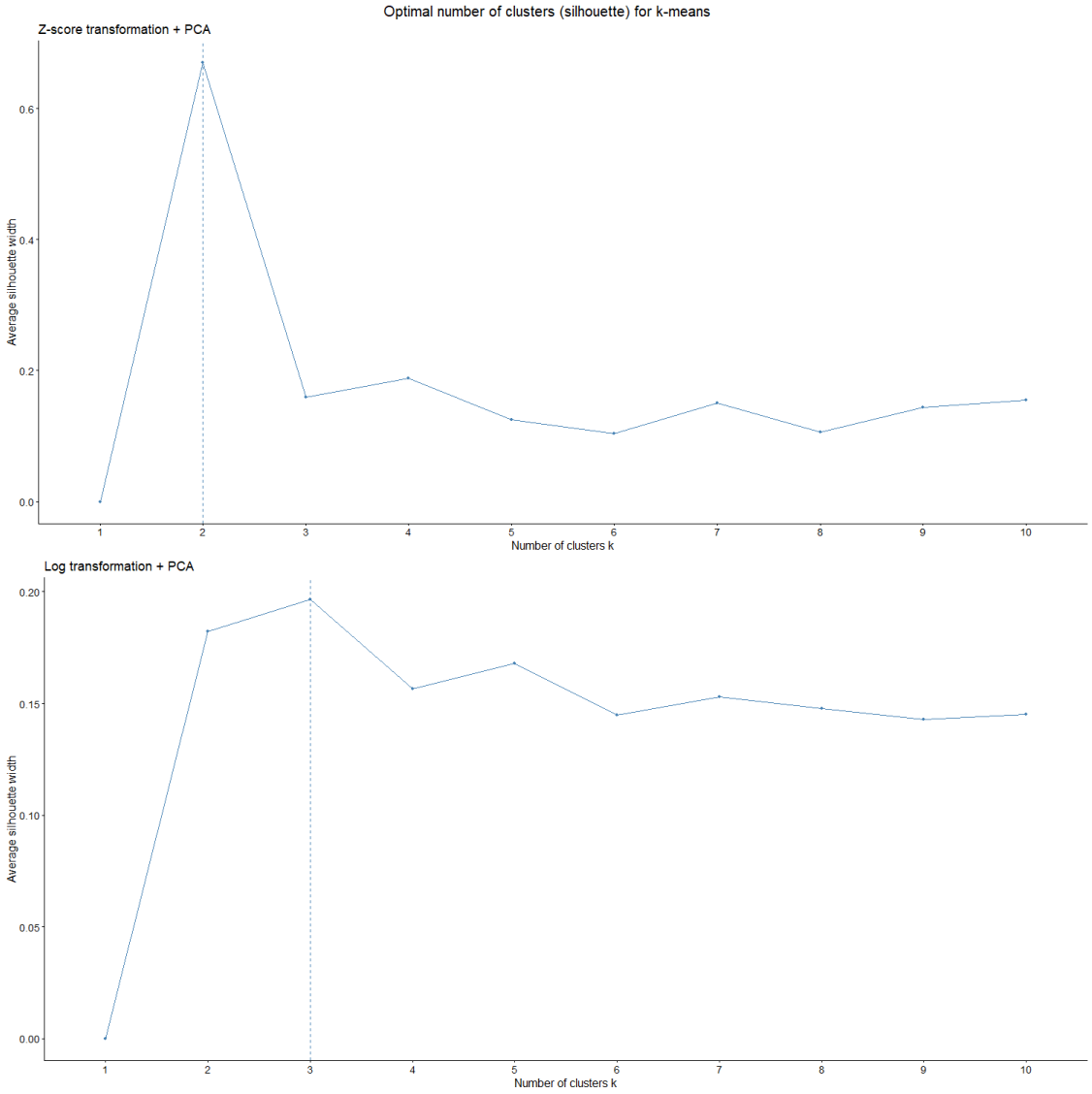

Figure 8: Number of clusters for k-means (with PCA)

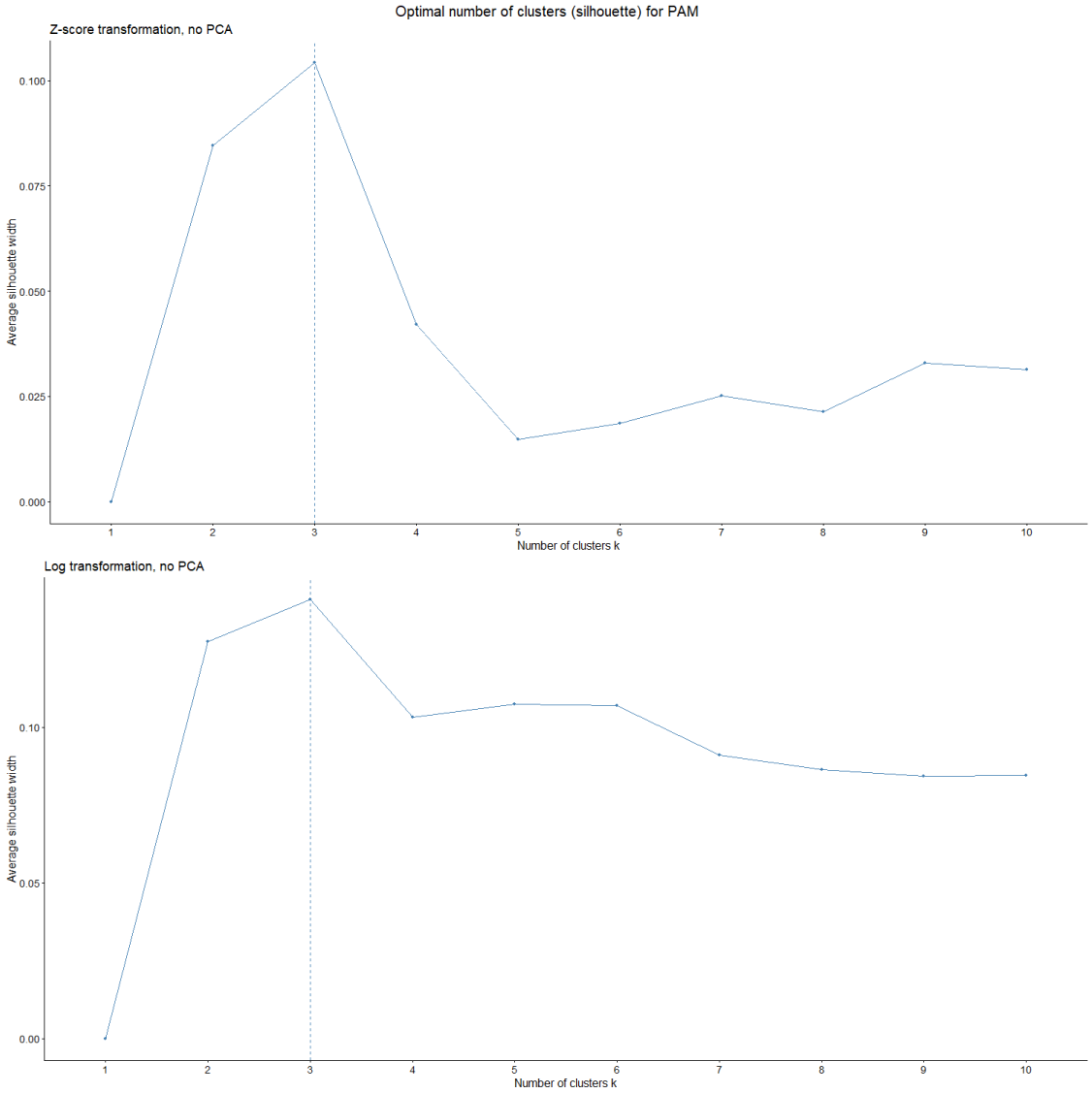

Figure 9: Number of clusters for PAM (no PCA)

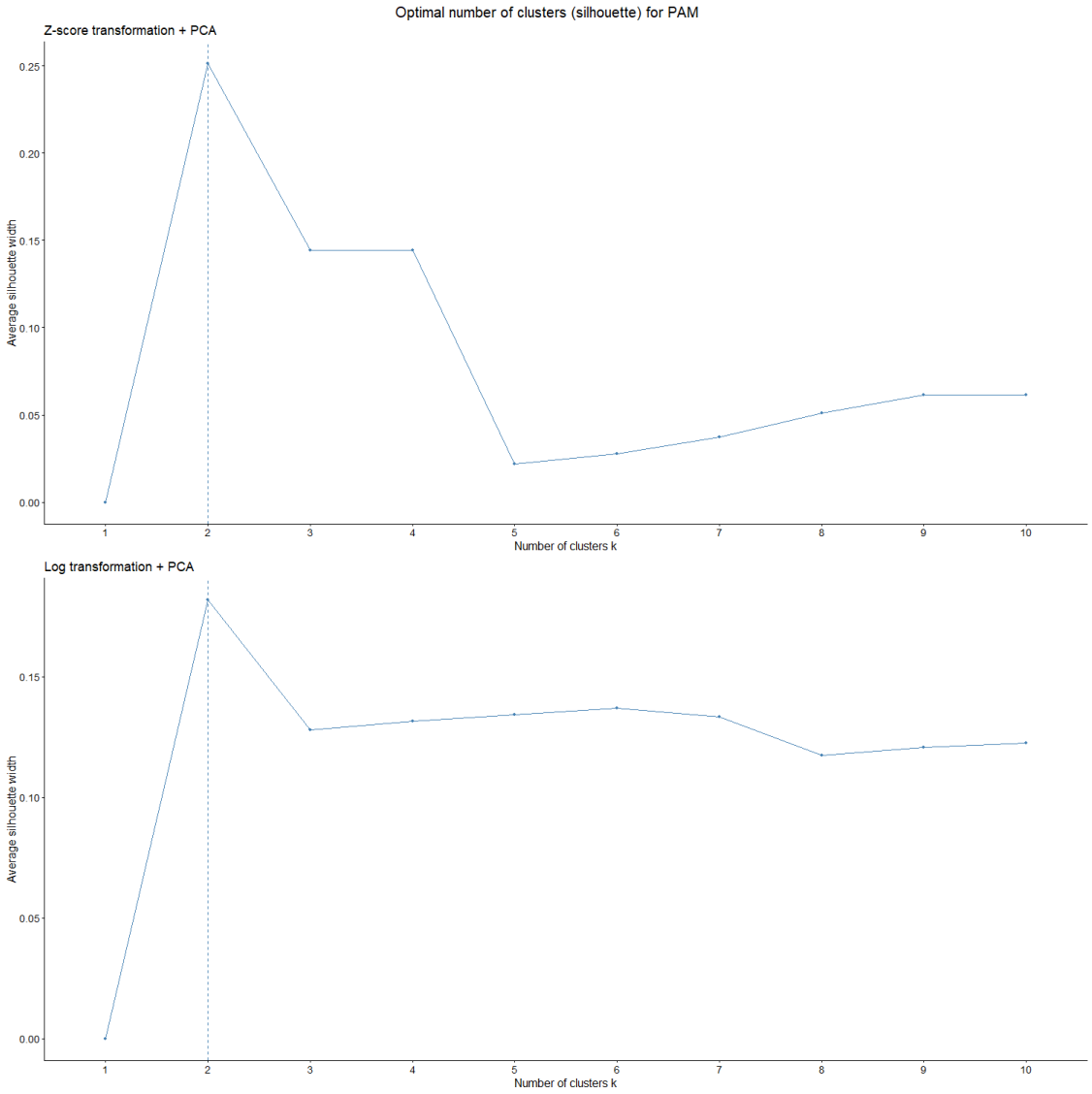

Figure 10: Number of clusters for PAM (with PCA)

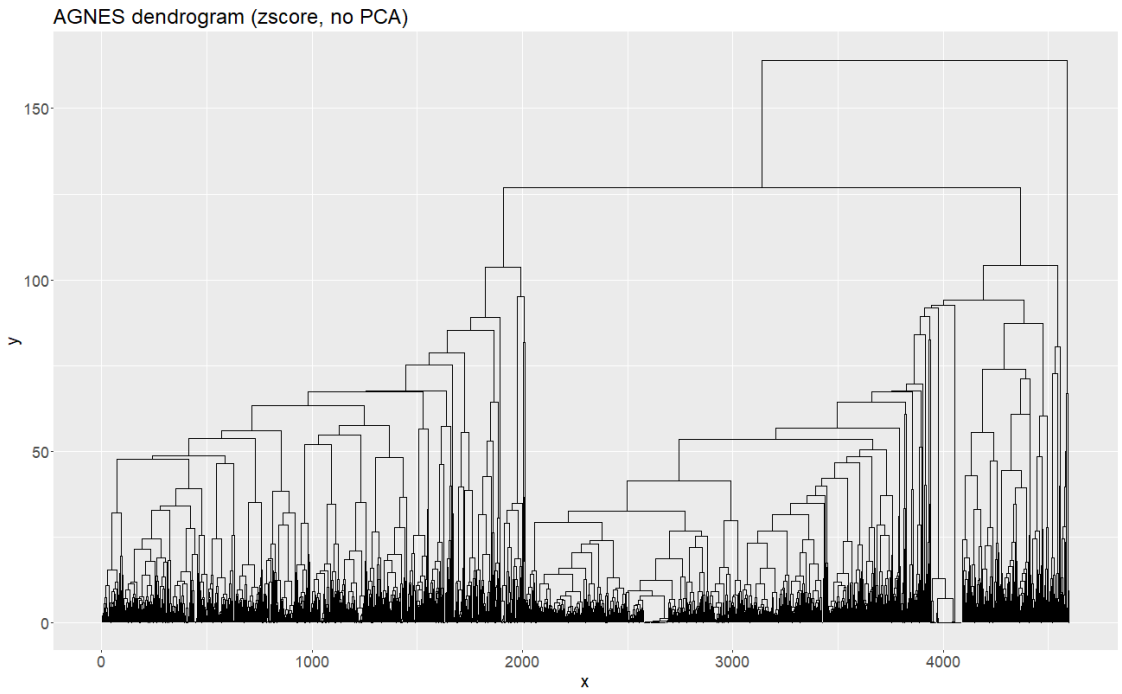

Figure 11: Number of clusters for AGNES (no PCA)

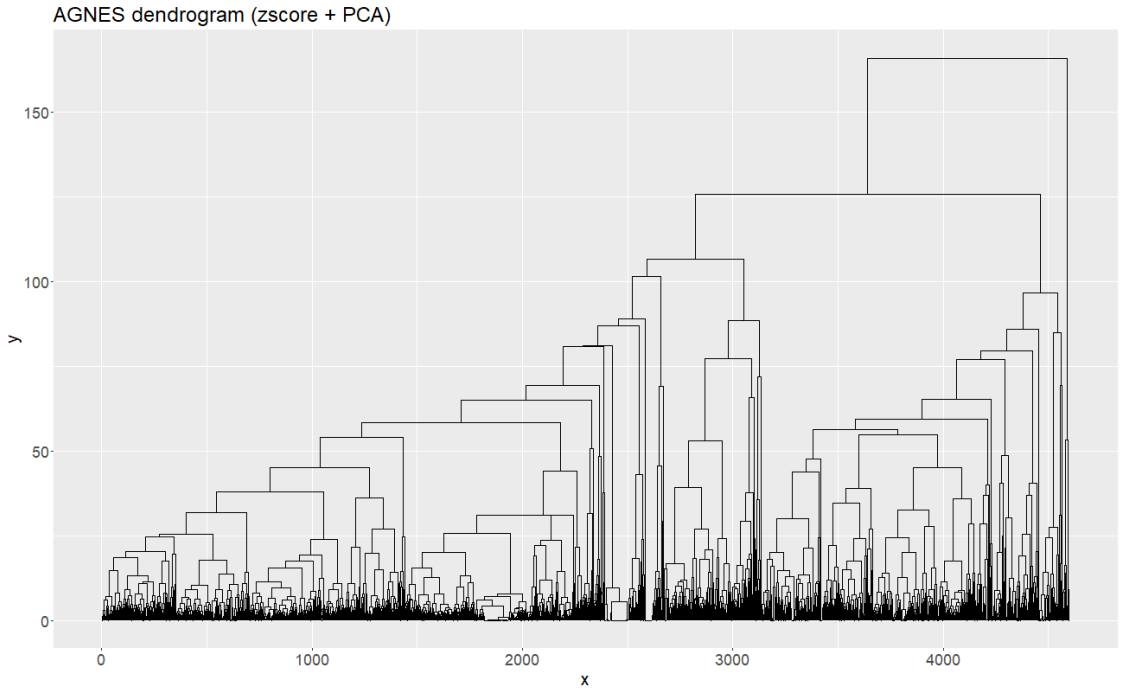

Figure 12: Number of clusters for AGNES (with PCA)

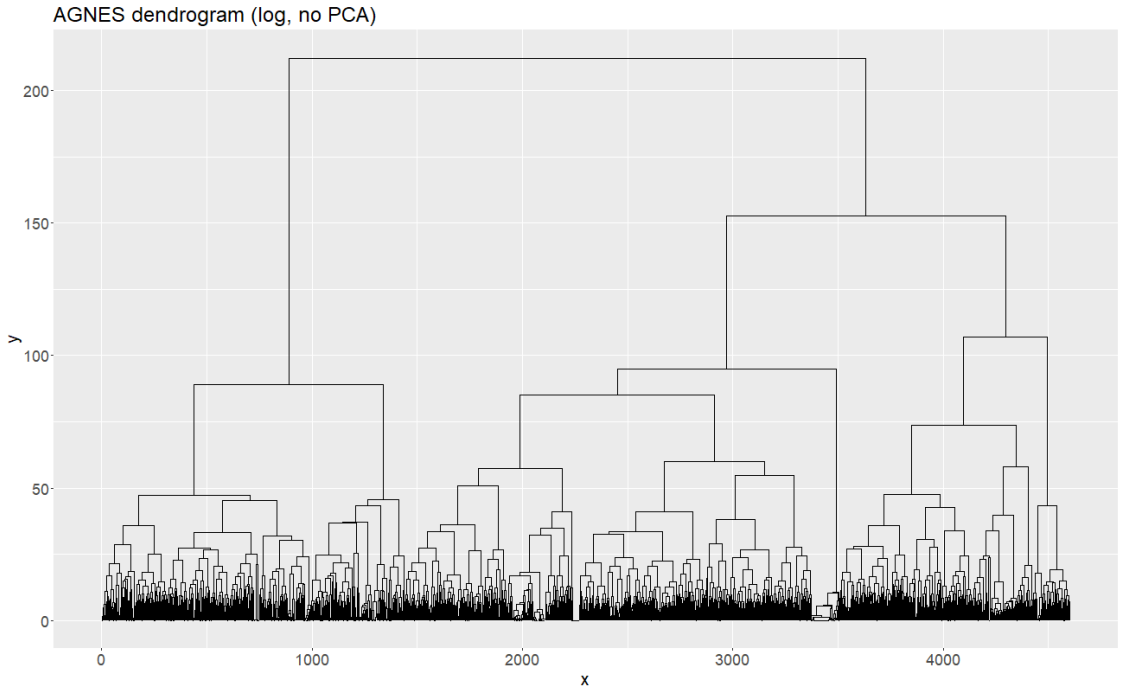

Figure 13: Number of clusters for AGNES (no PCA)

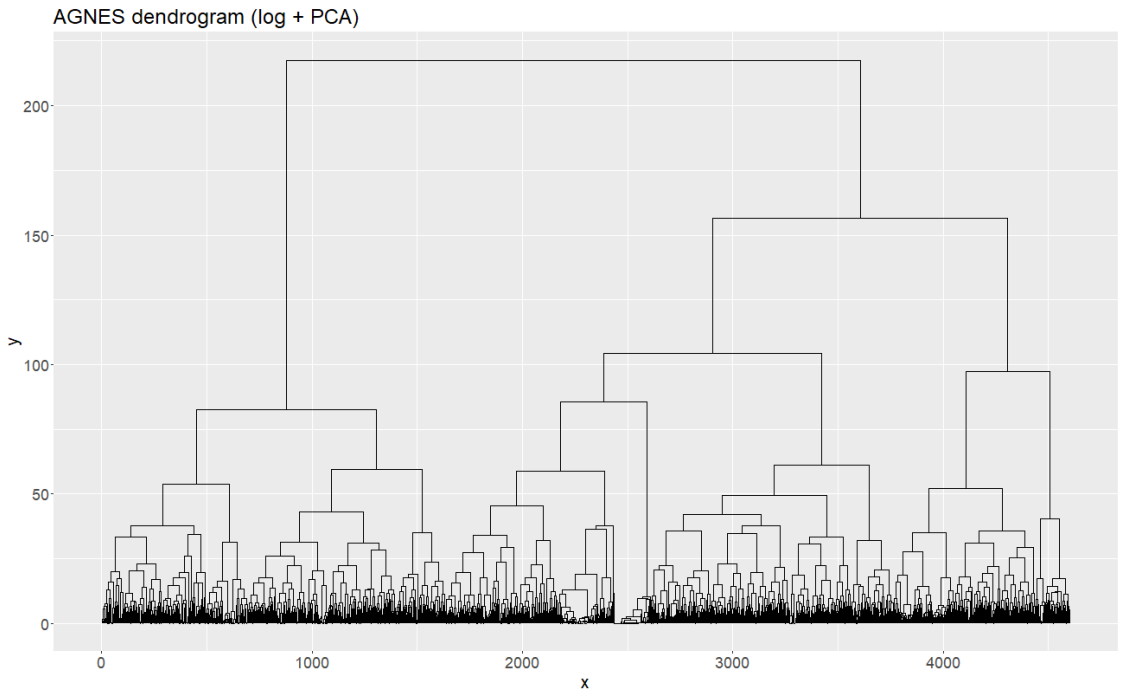

Figure 14: Number of clusters for AGNES (with PCA)

4.1 k-means

Figure 15 shows the results of clustering with k-means for z-score transformed data without PCA. We can see from this plot that cluster number 1 contains much more observations than the cluster number 2. Moreover, it looks like the nonspam class dominates in both of the clusters. We can verify this by looking at Table 1 – most observations from both classes fall into the first cluster, so this time the k-means algorithm wasn't able to differentiate well between spam and nonspam messages. What's interesting, applying PCA did not change the results much for the standardized data. Both Figure 16 and Table 2 display results almost identical to those observed in the previous case.

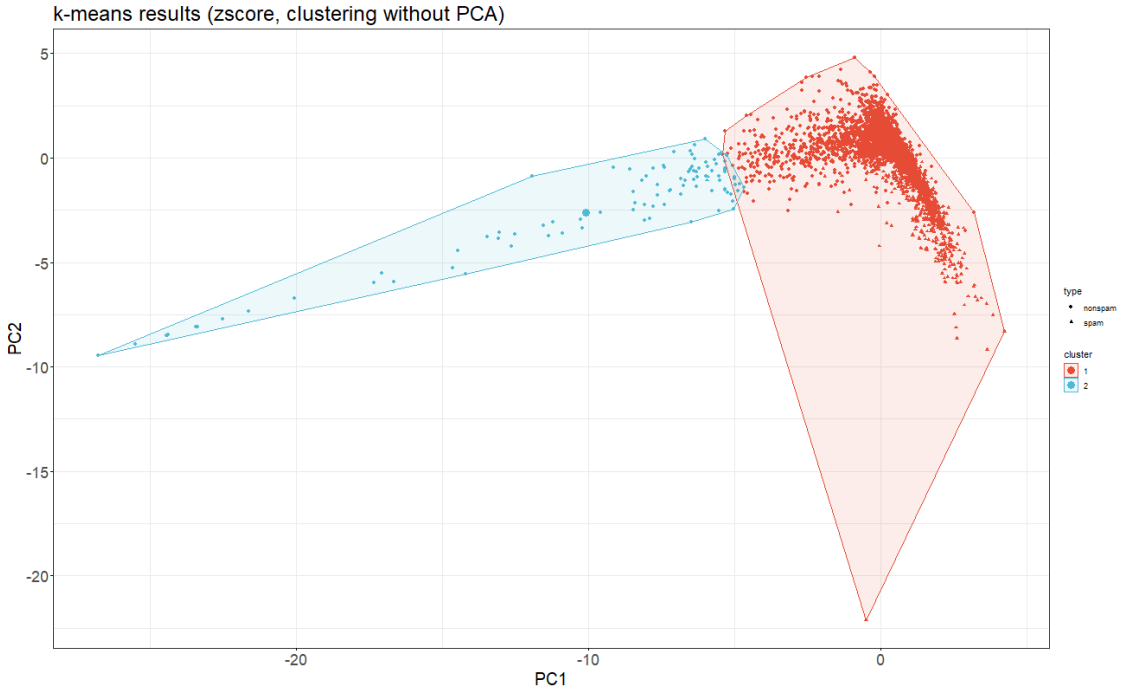

Figure 15: Clustering with k-means (z-score, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	2676	1811
2	112	2

Table 1: Clustering with k-means (z-score, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	2751	1813
2	37	0

Table 2: Clustering with k-means (z-score with PCA)

Situation looks quite different for the log-transformed data. Figure 17 shows cluster assignment for the no-PCA case. We see that the cluster number 3 contains mostly nonspam class, while the other two clusters consist mainly of spam, and Table 3 confirms that. Very similar situation is observed for log-transformed data with PCA. From Figure 18 we see that the cluster assignment didn't change much, except the names for cluster 1 and cluster 2 are swapped. Also looking at Table 4 we can say that the proportions of classes assigned to particular clusters is almost identical as in the previous case, so again we don't observe much influence on clustering from applying PCA to our data, but we can certainly say that it is much easier for k-means to differentiate between spam and nonspam in case of log-transformation rather than for the z-score scaling.

Now we will look at the internal cluster validation indices, namely Dunn index, silhouette width and connectivity. From Table 5 we can see that the largest value of Dunn index is obtained for z-score scaled data without PCA, while the lowest for log-transformed data with PCA. When it comes t silhouette width, the value closest to 1 is observed for the z-score transformed data with PCA, but z-score without PCA doesn't fall far behind. The lowest value of silhouette index results from clustering on log-transformed data

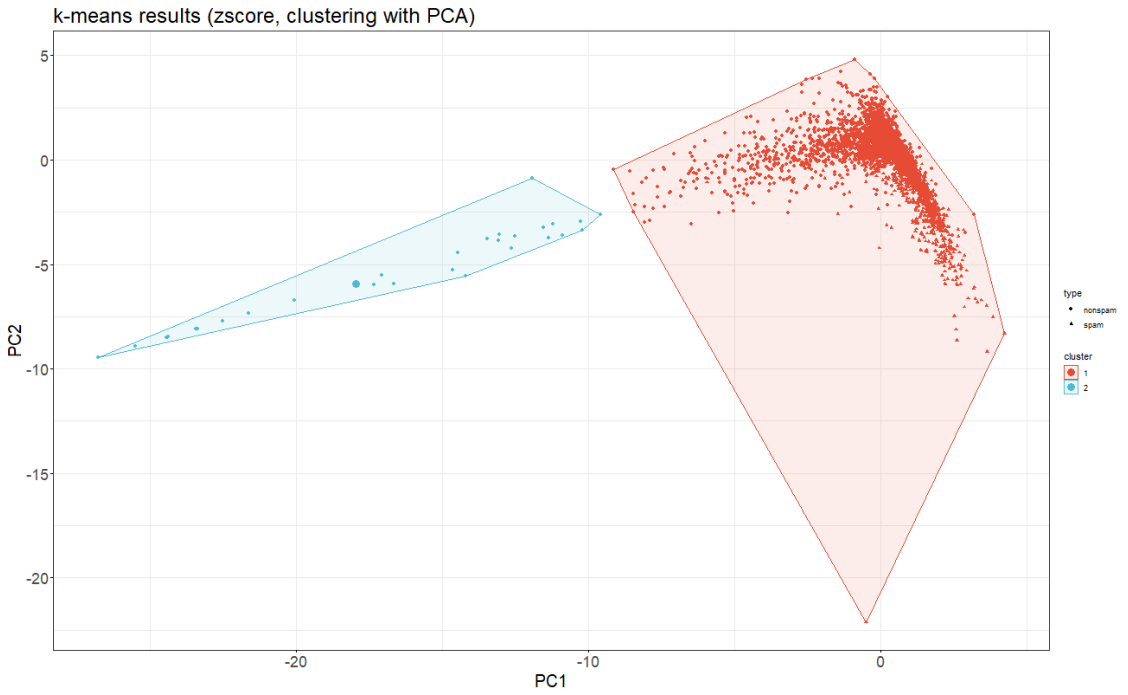

Figure 16: Clustering with k-means (z-score with PCA)

	Type	
Cluster	**nonspam**	**spam**
1	900	11
2	1626	326
3	262	1476

Table 3: Clustering with k-means (log, no PCA)

without PCA. Moreover, the connectivity for z-score scaled data, both with and without PCA has the same value, which is significantly smaller than the values obtained with log-transformed data. Considering those results, it seems that the k-means algorithm performs better with z-score than with log transformation, producing clusters of much higher quality.

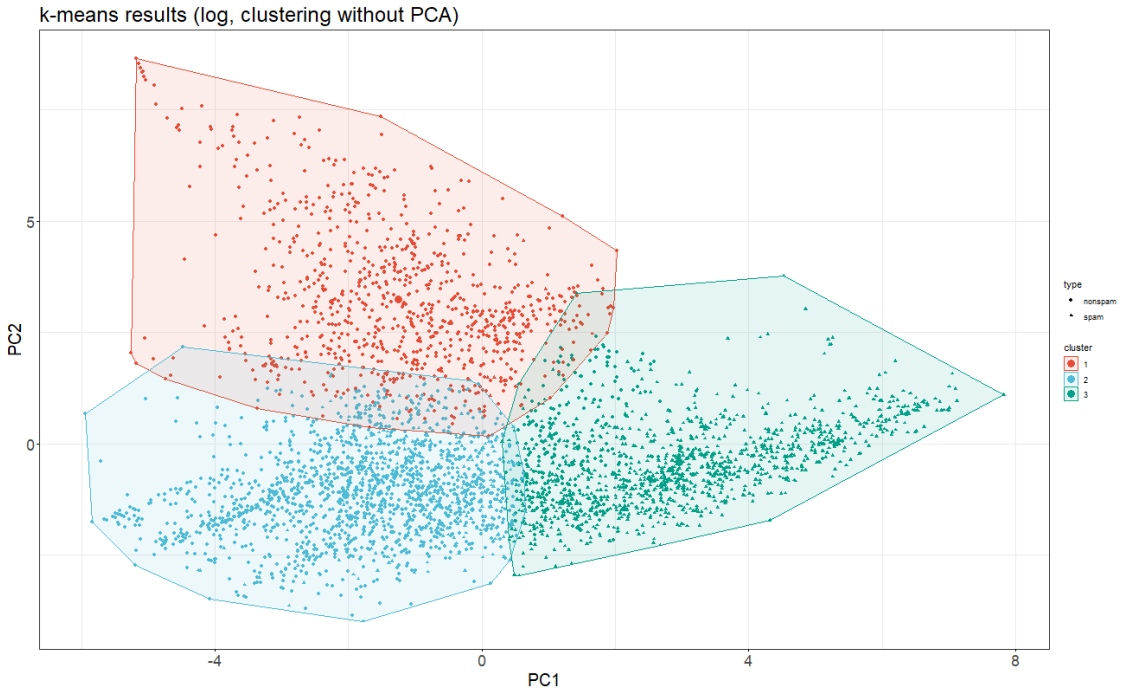

Figure 17: Clustering with k-means (log, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	1628	328
2	893	11
3	267	1474

Table 4: Clustering with k-means (log with PCA)

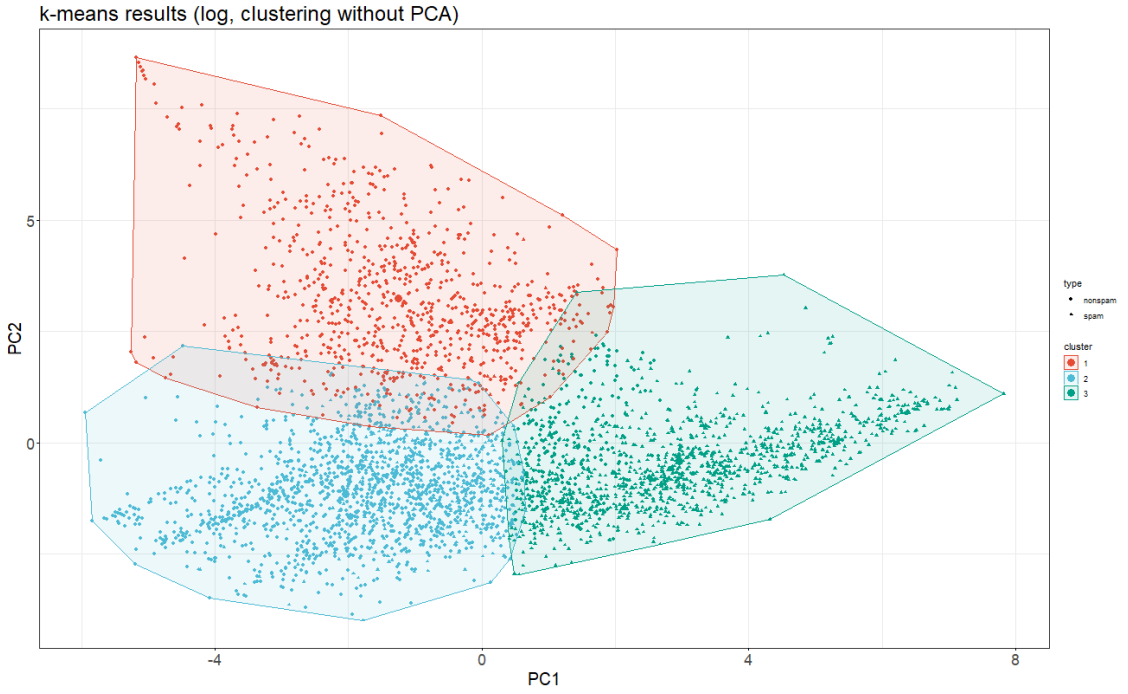

Figure 18: Clustering with k-means (log with PCA)

	Data transformation			
index	**zscore**	**log**	**zscore + PCA**	**log + PCA**
Dunn	0.9328	0.0187	0.7903	0.0177
Silhouette	0.8637	0.1513	0.8821	0.1965
Connectivity	2.9290	928.9421	2.9290	816.5504

Table 5: Internal cluster validation indices for k-means

4.2 PAM

For PAM method we again provide results for the same data transformations as for k-means. Figure 19 and Table 6 show results for standardized data. The situation, despite higher number of clusters is not significantly different that corresponding result for k-means algorithm. Clusters 2 and 3 contain almost only *nonspam* observations, but in cluster number 1 types are mixed. Let us now check the results obtained after applying PCA before PAM algorithm, that are shown on Figure 20 and Table 7. This time again the cluster number 1 contains observations with mixed types and the other cluster holds almost only *nonspam* observations.

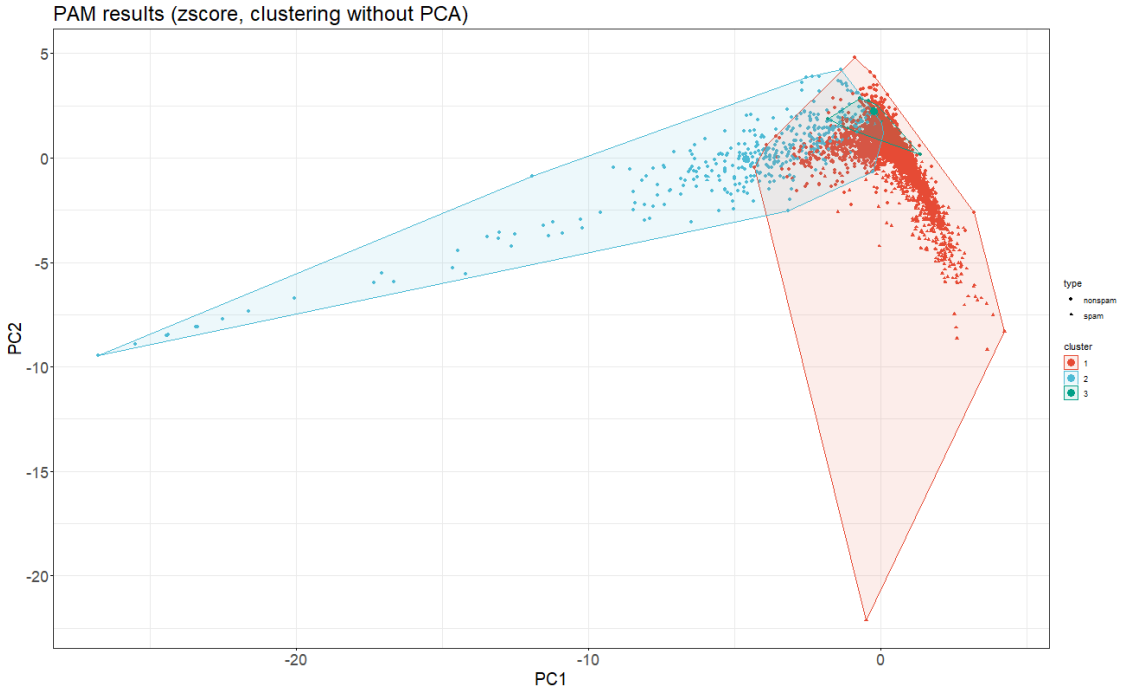

Figure 19: Clustering with PAM (z-score, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	2242	1810
2	421	3
2	125	0

Table 6: Clustering with PAM (z-score, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	2120	1798
2	668	15

Table 7: Clustering with PAM (z-score with PCA)

Next, we will use data after log-transformation, first results obtained from data with original dimensionality are presented on Figure 21 and Table 8. Once again results resemble the ones obtained with k-means algorithm. In one of the clusters majority of observations are of *nonspam* type, while the opposite is for second cluster. Results of PAM clustering after PCA are presented on Figure 22 and Table 9. Here we can observe notable difference. Here the observations of different types seem to be quite well separated between two created clusters. That is the case in which we observe the best partition when comparing to external indices. It also confirms that data transformations done before clustering have high impact on the result.

Table 10 shows the values of internal indices for PAM algorithm. This time we observe the best value of Dunn index for log-tranformed data without PCA, best silhouette width for z-score with PCA, and the smallest connectivity for z-score without PCA. In this case it is harder to distinguish which data transformation results with clusters of higher quality, but looking closely at the table, it seems that generally the z-score data performs slightly better. Moreover, we see that, on general, the results for PAM are worse than those for k-means.

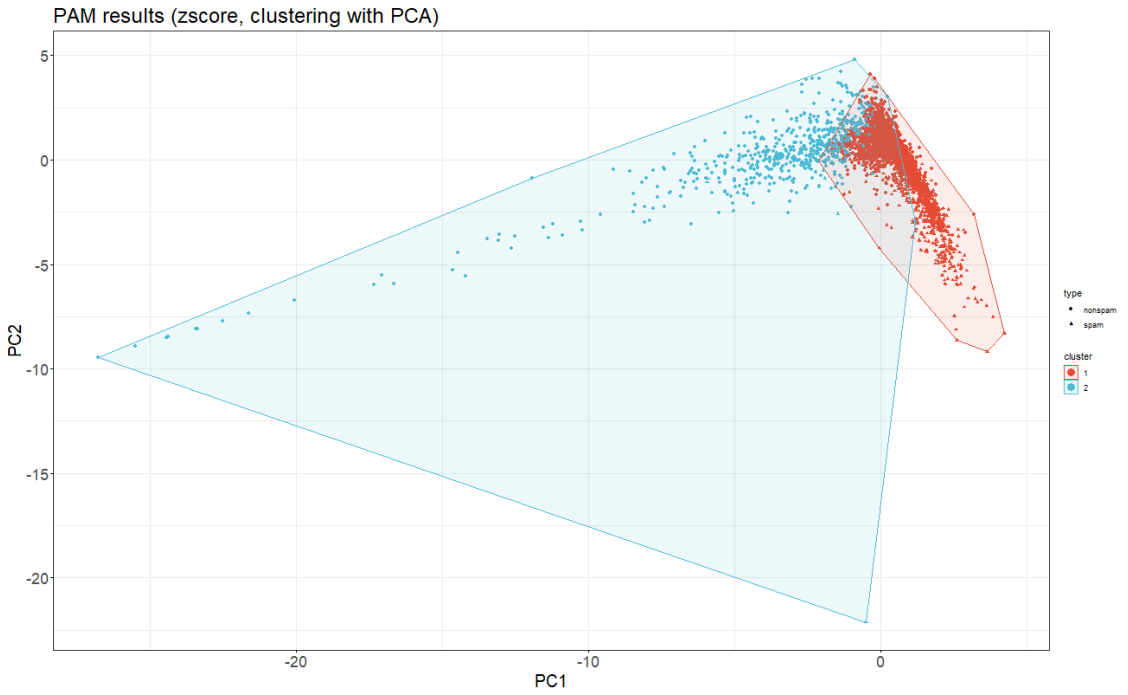

Figure 20: Clustering with PAM (z-score with PCA)

	Type	
Cluster	**nonspam**	**spam**
1	796	1558
2	1376	250
3	616	5

Table 8: Clustering with PAM (log, no PCA)

4.3 AGNES

The other type of clustering algorithm is the hierarchical clustering. We will take into consideration the agglomerative algorithm AGNES. The dendrograms were presented already in Section 4 as they were required to select the number of clusters that we want to split data to. To check how this clustering algorithm worked in case of *spam* and *nonspam* messages partition, we again provide similar plots. First for standardized data: Figure 23 and Table 11 show that all outliers placed in cluster 3 are *nonspam*. Other two clusters seem to separate two types of messages quite well, especially compared to previous results for this particular data. Let us now look, how AGNES algorithm will partition data after PCA, which is presented on Figure 24 and Table 12. Here the results are almost identical as without PCA, again one cluster contains *nonspam* outliers and the other two separate message types quite well.

We once more compare previous results to those based on log-transformed data for which results are introduced on Figure 25 and Table 13. This time, none of three created clusters contains outliers, however once again we notice that AGNES algorithm separated *spam* and *nonspam* messages noticeably well. Let us now check if this holds also for data after PCA, for which outputs are shown on Figure 26 and Table 14. There we can see that obtained results are analogous to the ones without PCA.

Let us now look at table 15, where results of internal cluster validation for AGNES are shown. Again, as in the previous cases, we observe that the log-transformed data produces clusters of poorer quality compared to z-score. For both approaches (PCA and no PCA) the higher Dunn index, higher silhouette width and lower connectivity are obtained for z-score. Considering the results obtained for all clustering algorithms, we can say that AGNES is the best performing one when it comes to cluster quality validated by internal indices.

5 Classification after applying PCA

For the last part of our project, we will perform the classification on log-transformed data after applying dimensionality reduction with PCA. We don't consider z-score scaled data, as in the previous part of the project we only ran the classification algorithms with the log transformation, and we want to compare the results obtained there (without feature selection) to those derived with the help of PCA.

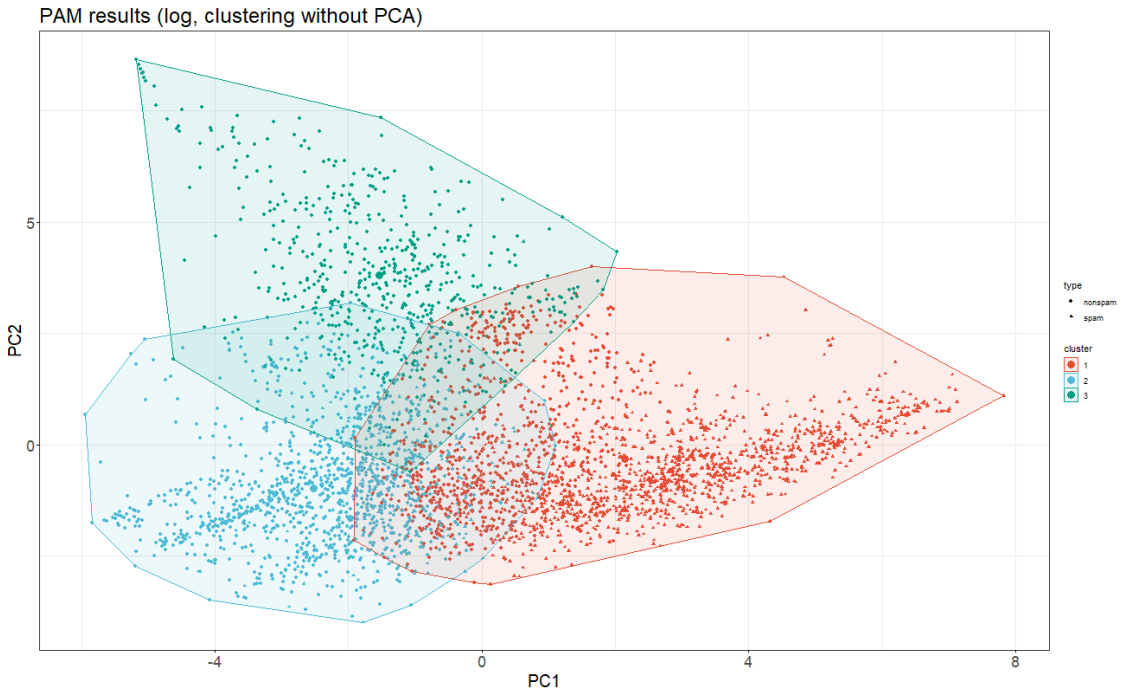

Figure 21: Clustering with PAM (log, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	203	1438
2	2585	375

Table 9: Clustering with PAM (log with PCA)

5.1 KNN

In Table 16 we can see the confusion matrix for KNN. The accuracy obtained here is 0.9383, which is almost the same as the one resulting from the original dataset (0.9391). In this case applying PCA algorithm didn't influence the results noticeably.

5.2 Random Forest

Let us now compare the results for the Random Forest classifier. From Table 17 we see that the accuracy here is 0.9478. This time we observe a very slight improvement (0.947 without PCA), however the difference is again not significant. As in the previous part of the project, the Random Forest performed better than KNN.

5.3 Discriminant Analysis

At the end, we will check how the results for LDA and QDA have changed after applying PCA. Tables 18 and 19 show the confusion matrices. The accuracy acquired for LDA in the previous part of the project was 0.9348, now we have 0.9243, so applying dimensionality reduction influenced the algorithm's performance badly. In case of QDA, the situation is opposite. Previously we had the accuracy of 0.8574, and now it is 0.8896. This is the only model where the PCA improved the accuracy noticeably, but as before, it is the worst performing classifier.

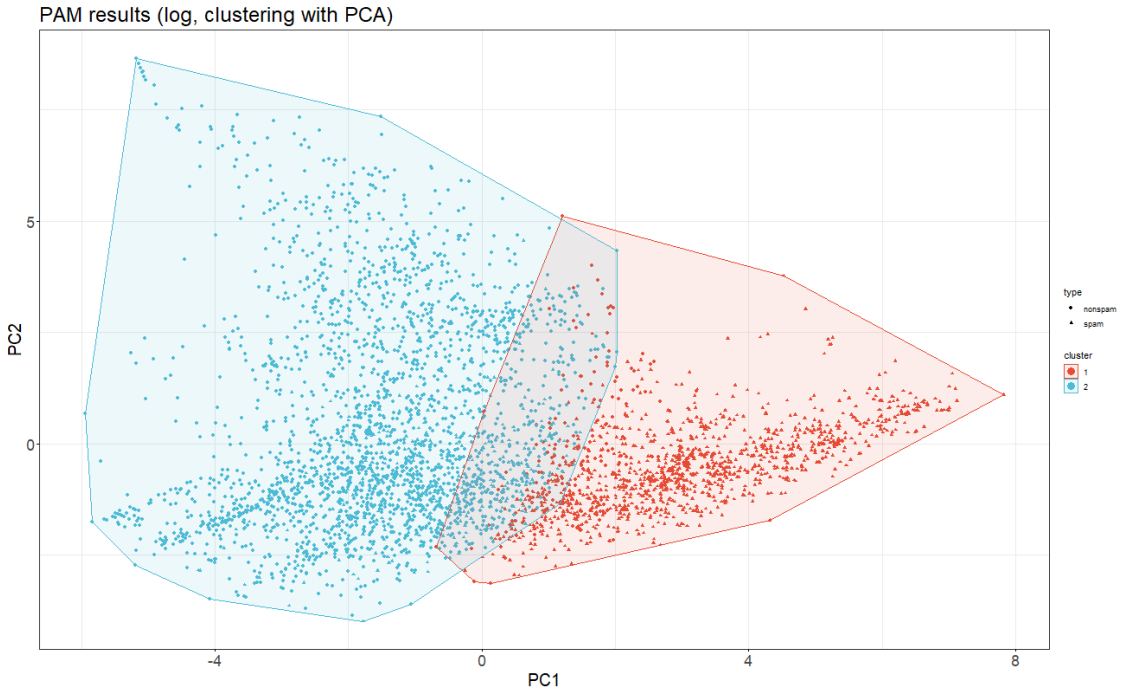

Figure 22: Clustering with PAM (log with PCA)

	Data transformation			
index	**zscore**	**log**	**zscore + PCA**	**log + PCA**
Dunn	0.0054	0.0092	0.0056	0.0008
Silhouette	0.1043	0.1411	0.2514	0.1818
Connectivity	383.9321	1385.6913	492.5373	637.2083

Table 10: Internal cluster validation indices for PAM

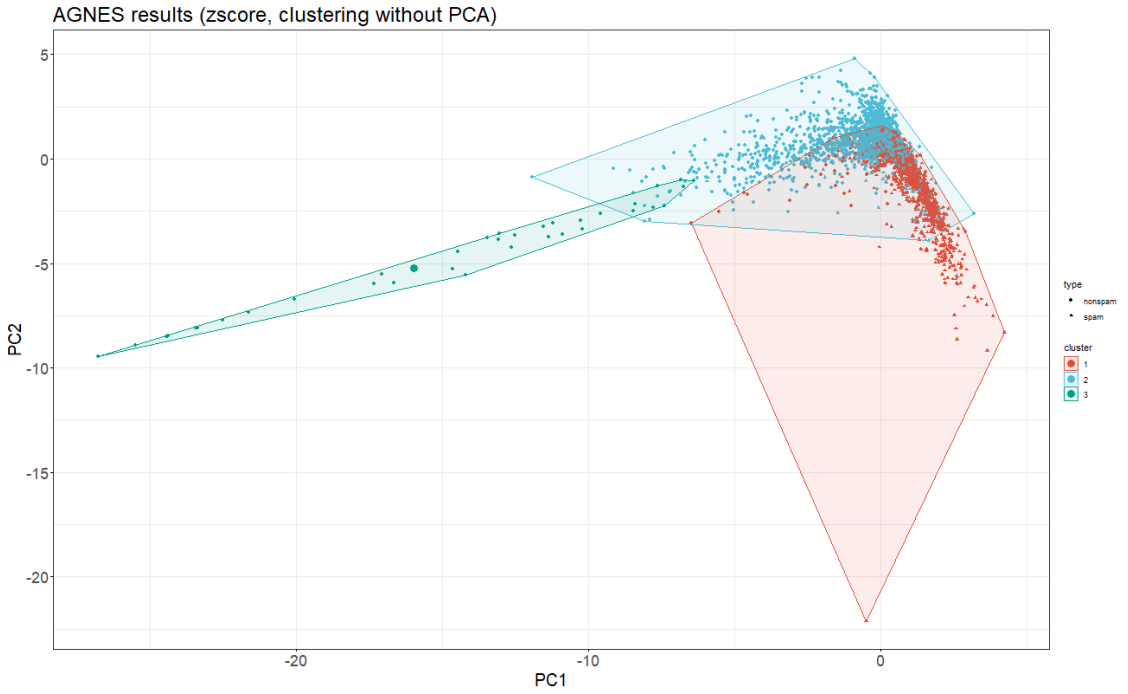

Figure 23: Clustering with AGNES (z-score, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	568	1445
2	2175	368
3	45	0

Table 11: Clustering with AGNES (z-score, no PCA)

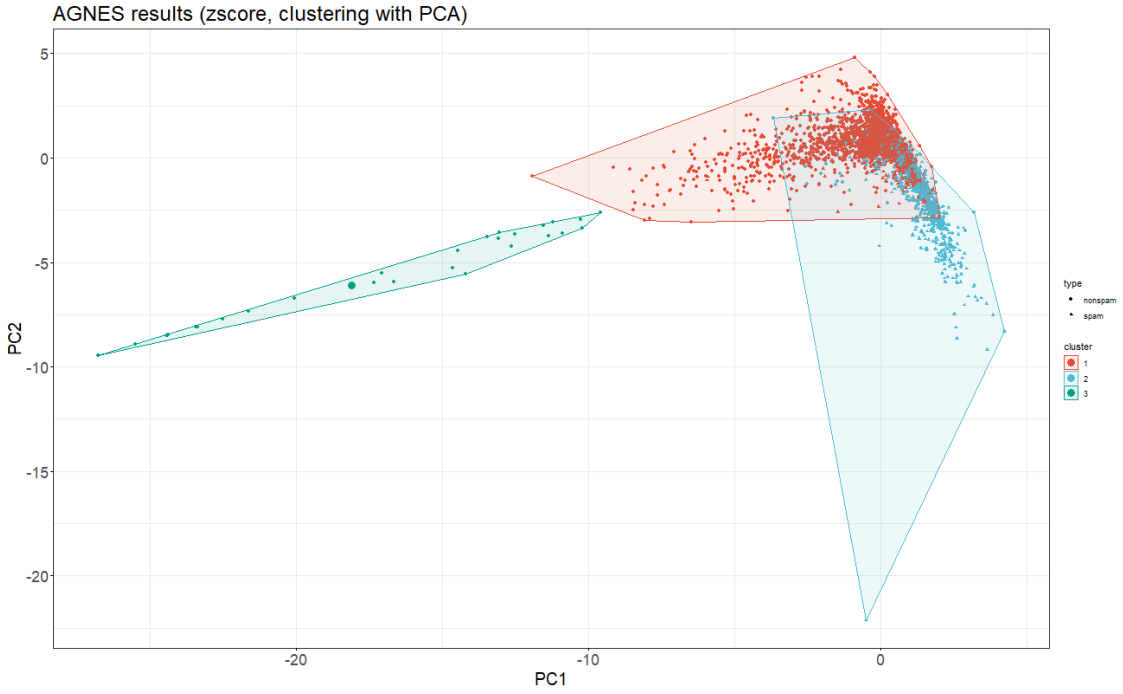

Figure 24: Clustering with AGNES (z-score with PCA)

	Type	
Cluster	**nonspam**	**spam**
1	2450	685
2	302	1128
3	36	0

Table 12: Clustering with AGNES (z-score with PCA)

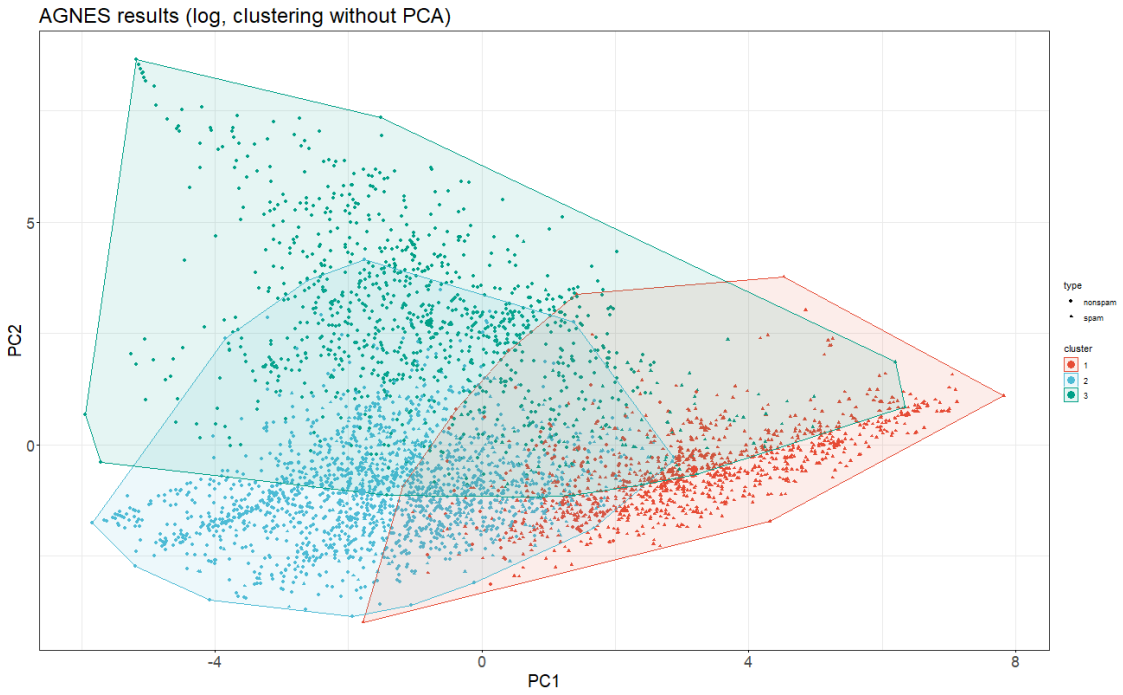

Figure 25: Clustering with AGNES (log, no PCA)

	Type	
Cluster	**nonspam**	**spam**
1	117	1322
2	1672	395
3	999	96

Table 13: Clustering with AGNES (log, no PCA)

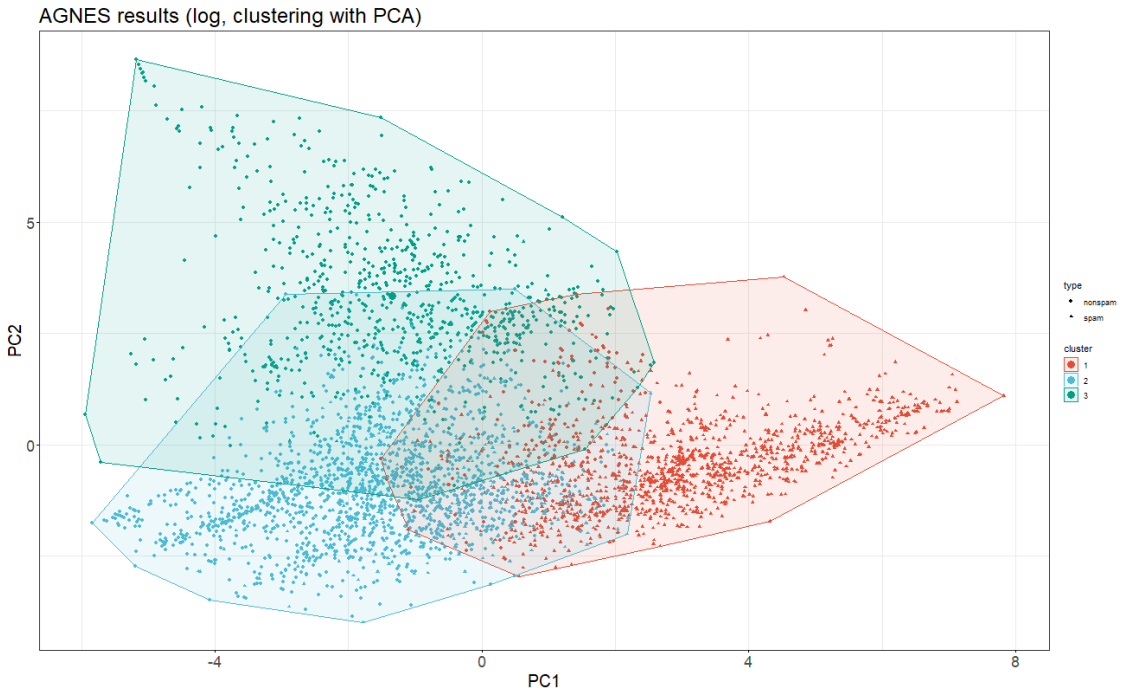

Figure 26: Clustering with AGNES (log with PCA)

	Type	
Cluster	**nonspam**	**spam**
1	196	1436
2	1765	367
3	827	10

Table 14: Clustering with AGNES (log with PCA)

	Data transformation			
index	**zscore**	**log**	**zscore + PCA**	**log + PCA**
Dunn	0.9328	0.4316	0.7903	0.1510
Silhouette	0.8637	0.2752	0.8821	0.2343
Connectivity	2.9290	5.8579	2.9290	77.0813

Table 15: Internal cluster validation indices for AGNES

	Reference	
Prediction	nonspam	spam
nonspam	649	23
spam	48	430

Table 16: Confusion matrix - KNN

	Reference	
Prediction	nonspam	spam
nonspam	665	28
spam	32	425

Table 17: Confusion matrix - Random Forest

	Reference	
Prediction	nonspam	spam
nonspam	666	56
spam	31	397

Table 18: Confusion matrix - LDA

	Reference	
Prediction	nonspam	spam
nonspam	591	21
spam	106	432

Table 19: Confusion matrix - QDA

6 Summary

The project focused on utilizing cluster analysis in combination with dimensionality reduction techniques on the *spambase* dataset. The objective was to evaluate the performance of clustering algorithms both in distinguishing between spam and non-spam emails and the quality of obtained clusters. Three clustering algorithms, namely k-means, PAM, and AGNES, were employed, along with PCA for dimensionality reduction. The project also compared classification results before and after applying dimensionality reduction. PCA also improved data visualization.

For clustering, the optimal number of clusters was determined using silhouette index for k-means and PAM, and dendrogram analysis for AGNES. Results showed varying optimal cluster numbers depending on the algorithm and data transformation.

The clustering results were analyzed for each algorithm and transformation. For k-means, clusters tend to favor one class over the other, with log transformation help in better differentiation between *spam* and *nonspam*. PAM results resembled k-means, while AGNES showed better separation between message classes, especially with z-score scaling.

Internal cluster validation indices were computed, with all algorithms performing better with z-score scaling, and AGNES showing superior performance overall.

Finally, classification was performed after applying PCA. Results indicate minimal impact on KNN, Random Forest and LDA classifiers, while QDA's accuracy improved noticeably after dimensionality reduction.

Overall, the project demonstrated the importance of data preprocessing, choice of clustering algorithms, and the impact of dimensionality reduction techniques on clustering algorithms and classification methods performance.