CLUSTERING COMPANIES USING CYBER SECURITY METRICS, CATEGORY VARIABLES, AND TEXT

Harris Asad

Supervisor: Dr. Eleni Matechou

Submitted in partial fulfilment of the requirements
for the degree of **MSc Statistical Data Science**
University of Kent, 2021

Acknowledgments

I would first like to thank all the professors and staff members at the school of Mathematics Statistics and Actuarial Sciences (SMSAS), University of Kent, for their immense support and resilience to teach and help us during this incredibly tough year of online learning. I would also like to thank my supervisor for checking up on me from time to time and keeping me on track throughout the course of this project. Finally I would like to thank Mr. William Hesselmann and Mr. Luke Wilson Mawer from KYND (PVT.) LTD. for providing us the data of cyber security metrics for different companies and properly guiding us on what they expect from this project.

Contents

List of Figures

List of Tables

Declaration of Originality

I hereby declare that the work contained in this report and the intellectual content of it is the product of my own work. This report has not been previously published in any form nor does it contain any verbatim of the published resources which could be treated as infringement of the international copyright law. I also declare that I do understand the terms 'copyright' and 'plagiarism,' and that in case of any copyright violation or plagiarism found in this work, I will be held fully responsible of the consequences of any such violation.

(Harris Asad)
September, 2021
University of Kent

Abstract

This project features analysing cyber security metrics provided by KYND (PVT.) LTD. The objective of the project was to cluster companies based on these metrics and find out what factors lead to more risks of any type. After some data preparation, the dimensions of our data were first reduced with a principal component analysis (PCA) and feature contrasts were established. It was found out that high number of e-mails and certificate risk metrics lead to low services risk metrics and vice versa. Similarly high number of e-mail risk metrics lead to less certificate risk metrics and vice versa. Companies showing such behaviours were then seen forming clusters when K-means clustering was applied to our data. The companies showing such behaviours were low compared to the overall data since the majority of our data points had low scores in our PCA. Hierarchical clustering was also applied to our data which did not amount to any relevant insight and could not produce any comparable result to our K-means clustering technique.

Introduction

1.1 What is KYND?

This project is in collaboration with **KYND (PVT. LTD.)**, a startup based in London. KYND is a cyber risk management software product which measures cyber risk exposures to a company's website and provides rectification guidelines for them. The software uses a simple traffic light system to cluster cyber risk exposures based on their severity to either of three colors i.e. Red, Amber or Green. The company calls them RAG. The software at the end then publishes a report of these risks and provides remedial measures to address them. The following figures provide an example simulation of how the software works for a company called squareball:

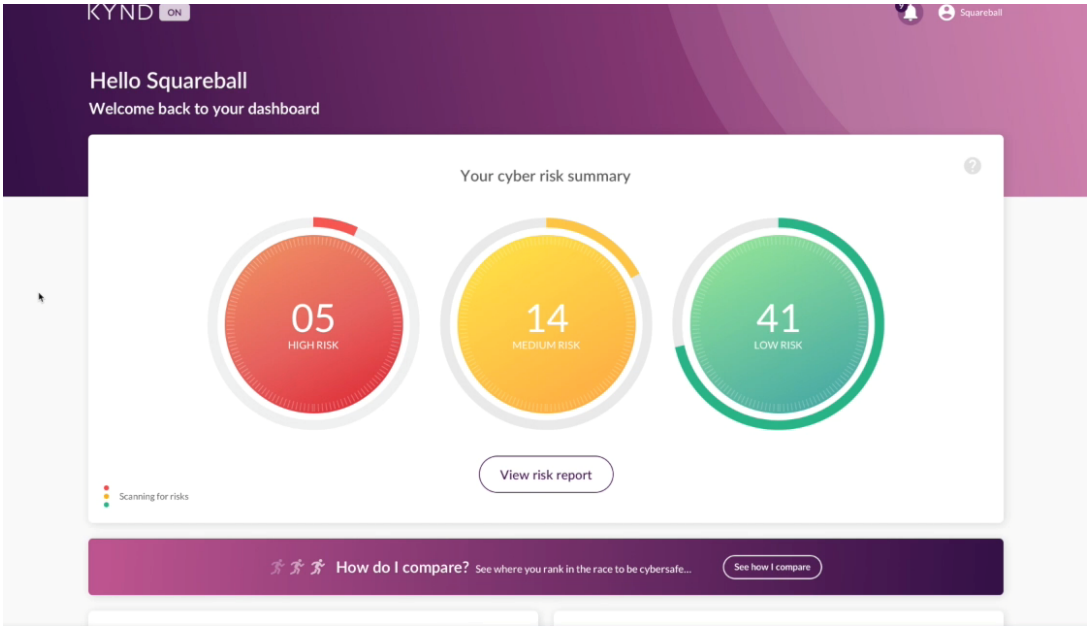

Figure 1.1: Cyber risks clustered based on RAG after scanning the website

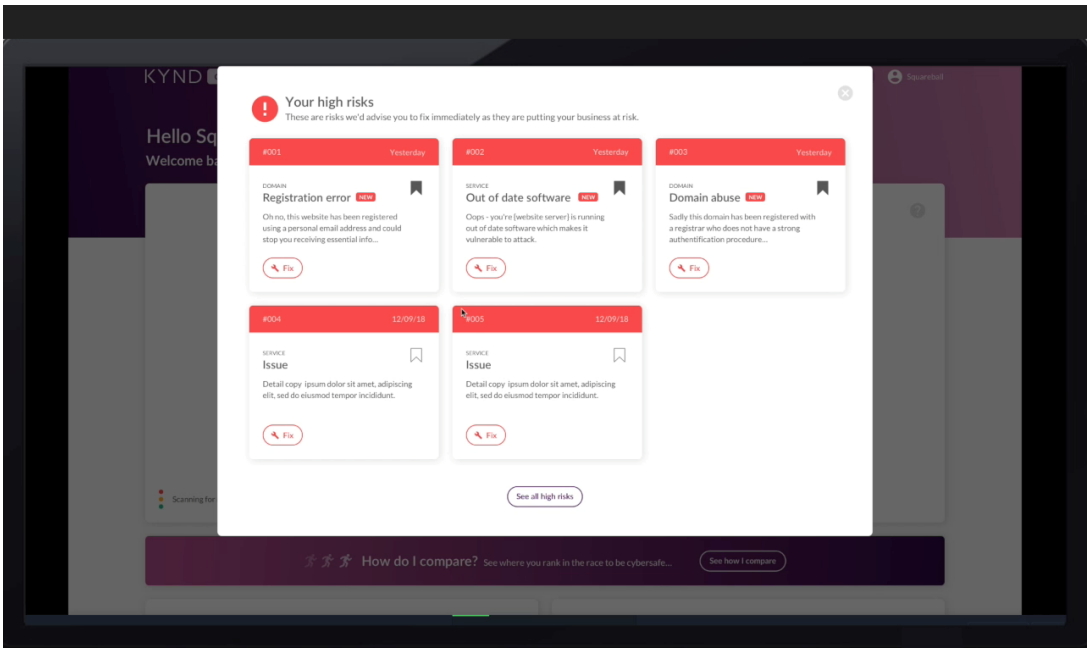

Figure 1.2: The five red risks highlighted and their rectification offered

1.2 Cyber Risks Measured by KYND

To provide some context on what types of cyber risk exposures are picked up by the KYND software, we need to list them accompanied by their short description. This information is necessary when making sense of our data analysis later on.

Table 1.1: Types of Cyber Risks

Risk	Description
Certificates	This is basically a security certificate which, if up to date, ensures that the website is providing a secure environment for the user to carry out transactions.
Services	This includes internet services running on the website which maybe at the risk of, for example, being out of date or misconfigured.
E-mail	This includes all risks related to the company's e-mail for example e-mail spoofing or spamming .
Phishing	Phishing occurs when an online attacker acting as a trusted organization/person frauds someone into giving up sensitive information like credit card information, login credentials etc.
Ransomware	Ransomware is a malware that keeps a system hostage and prevents access to it until some kind of ransom is paid to it.

Domain	A domain is the name of the website. The software tracks down all the web domains owned by the company and measures risks related to them.
Credentials	This includes all tools for authentication and verification of a persons identity for example login username and password

It is to be noted that all above risks, if any, are clustered under the RAG classification based on their severity as discussed earlier.

1.3 Data Provided by KYND

1.3.1 Cyber Risk Metrics Data

KYND provided us data for 910 companies along with their cyber-security metrics. The database consists of 44 columns, the summary of which is provided below:

Table 1.2: Cyber Risk Metrics Data

Column Name	Data Type	Description
orgId	Character	Unique ID of the company for reference
orgName	Character	Name of the company e.g Vipera Limited
Domain	Character	This is the URL of the company's website e.g. vipera.com
userId	Character	Unique ID of the user. This is same as orgId. Was told to neglect this column.
servicesCount	Numeric	Number of services running on the website
sldCount	Numeric	Number of domains found for this organization.
certificates_red	Numeric	Number of red risks for the certificates.
certificates_amber	Numeric	Number of amber risks for the certificates.
certificates_green	Numeric	Number of green risks for the certificates.
credentials_red	Numeric	Number of red risks for credentials.
credentials_amber	Numeric	Number of amber risks for credentials.
credentials_green	Numeric	Number of green risks for credentials.
emails_red	Numeric	Number of red risks for e-mails.
emails_amber	Numeric	Number of amber risks for e-mails.
emails_green	Numeric	Number of green risks for e-mails.

phishing_red	Numeric	Number of red risks for the phishing.
phishing_amber	Numeric	Number of amber risks for the phishing.
phishing_green	Numeric	Number of green risks for the phishing.
ransomware_red	Numeric	Number of red risks for the ransomware.
ransomware_amber	Numeric	Number of amber risks for the ransomware.
ransomware_green	Numeric	Number of green risks for the ransomware.
services_red	Numeric	Number of red risks for the services
services_amber	Numeric	Number of amber risks for the services.
services_green	Numeric	Number of green risks for the services.
slds_red	Numeric	Number of red risks for domains.
slds_amber	Numeric	Number of amber risks for domains.
slds_green	Numeric	Number of green risks for domains.
subdomainCount	Numeric	Number of subdomains found in the website
domains_numerator	Numeric	This is the numerator value of total number of hidden domains.
domains_denominator	Numeric	This is the denominator value of total number of domain count for the organization. This column's value is similar to sldCount column.
domains_percentage	Numeric	This is percentage of number of hidden domains
domains_classification	Character	High percentage of hidden domains is classified as "straggling", medium as "off the back" and low as "In the group".
mis_services_numerator	Numeric	This is total number of misconfigured services found on the website as numerator.
mis_services_denominator	Numeric	This is total number of all services found on the website as denominator. This column is similar in value to servicesCount column.
mis_services_percentage	Numeric	This is percentage of number of misconfigured services
mis_services_classification	Character	High percentage of misconfigured services is classified as "straggling", medium as "off the back" and low as "In the group".
ood_services_numerator	Numeric	This is total number of out of date services as the numerator.

ood_services_denominator	Numeric	This is total number of all services found on the website as denominator. This column is similar in value to servicesCount column.
ood_services_percentage	Numeric	This is percentage of number of out of date services.
ood_services_classification	Character	High percentage of out of date services is classified as "straggling", medium as "off the back" and low as "In the group".
certificates_numerator	Numeric	This is total number of expired certificates as numerator.
certificates_denominator	Numeric	This is total number of all certificates for the website as denominator.
certificates_percentage	Numeric	This is percentage of number of expired certificates.
certificates_classification	Character	High percentage of expired certificates is classified as "straggling", medium as "off the back" and low as "In the group".

It is also to be noted that for each classification category, what counts as high, medium or low percentage is pretty flexible and is entirely dependent on what type of quantity we are looking at. For example the value of 40% for percentage of *out of date services* for an organization may count as high percentage in this category, but medium for percentage of *expired certificates category.*

1.4 Objectives of The Project

The objectives of the project are two-fold:

- To cluster companies based on the provided cyber risk metrics i.e. what factors lead to more red, amber and green risks of any type.
- To find similarities and differences between companies and grouping them by referring to these risk metrics.

These objectives are based on communication with KYND as to what they expect on the type of information to be extracted from this dataset.

1.5 Methodology to be Applied to the Data

The objectives show that this project will require clustering the companies based on these cyber risk metrics. This seems like a textbook case of using *unsupervised learning techniques* to subgroup the data. This is because we neither have a particular response variable to our data nor are we interested in any type of prediction, instead we have a set of features upon which we intend to find interesting insights. [1]

The two unsupervised learning techniques encountered during our course which we will apply to our data are as follows:

1. Principal Component Analysis (PCA)
2. Clustering
   (a) K-means Clustering
   (b) Hierarchical Clustering

We will first prepare the data for this multivariate data analysis by analysing which features to include and exclude in our analysis. This will be followed by reducing dimensions of the dataset to a set of principal components that account for maximum variation in the data. These principal components will then be used to graphically visualize any clustering of the companies after applying K-means Clustering. Hierarchical Clustering will then be used to further analyse any subgroups within our data and if they corroborate our conclusions from the K-means Clustering technique.

For our whole data analysis practice, we will be using *R Studio*.

Data Analysis

2.1 Data Preparation

The summary statistics for our data after importing it to R studio showed that some features in our data set are constant or have null value throughout. These include the credentials columns for all colors, phishing columns for red and amber colors and ransomware columns in all colors. They must be removed as they will not show any variance and will hamper our analysis by not being able to get the covariance matrix and the correlation matrix. Similarly some features are also not relevant to our analysis like organisation ID, domain name, user ID and organisation name. Instead it will be much easier to recognise companies if they are numbered starting from 1 for our analysis.

We can also safely remove the numerator and denominator values for all cyber metrics as the percentage values will be enough to get the trend, and also because the denominator values are the same as some other features in our dataset (refer to Table-2). Similarly the classification done is based upon our percentage value, so removing it to have all our features in numeric form will simplify our analysis as the percentage field for all our cyber security metrics will be enough to graphically visualize its affect on other metrics.

After performing all this operation, we are left with 21 columns, with the first column referring to the row number (910 in total) and it is named domain. This is basically the ID of our companies starting from 1, which will make it easier for us to identify them on a plot. To perform our PCA to the data, we first need to see if our remaining features are correlated, otherwise simplification with highly uncorrelated features will not lead to any simplification [2]. We get the following correlation plot:

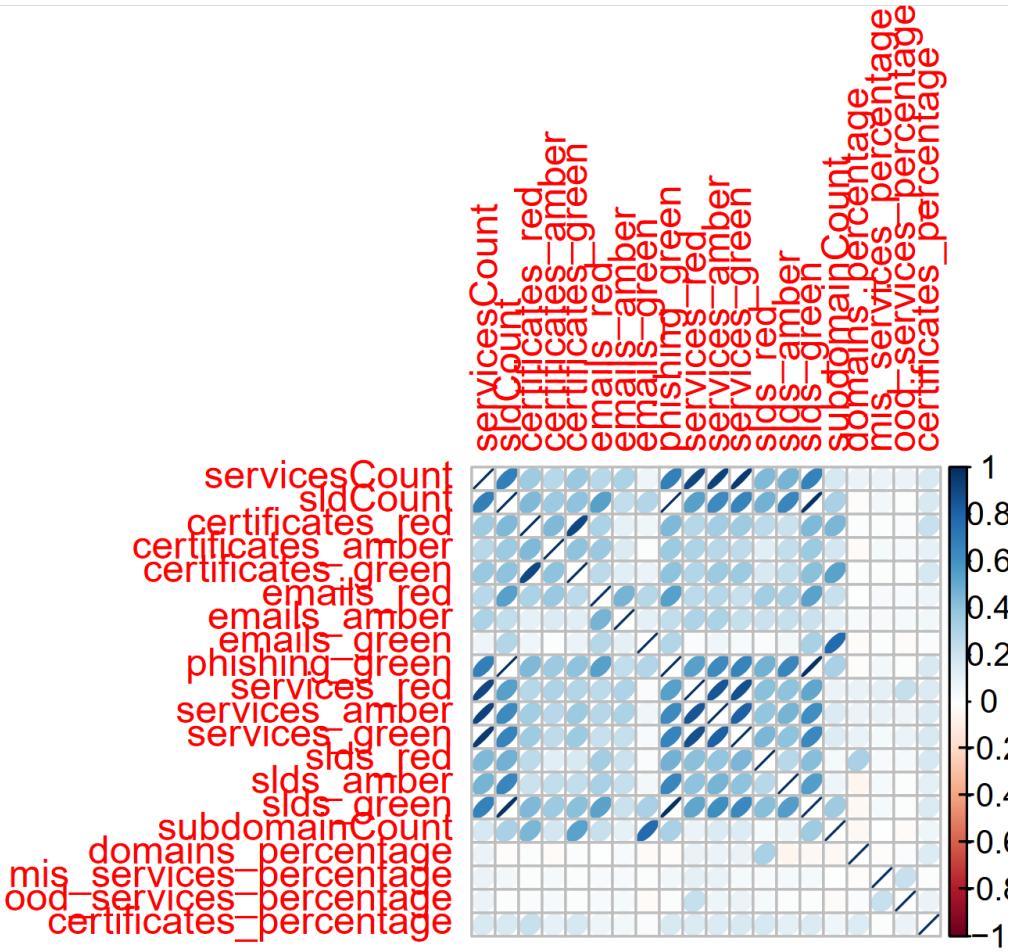

Figure 2.1: Correlation plot for our remaining dataset

The majority of our variables are positively correlated especially the services and certificates metrics with each other. One question my arise that proportions of these two metrics should then be used instead for each color, but when we divide by the total number, we get the same ratio for all three colors as before and surely after assigning proportions to our metrics we get the same correlation plot as above. We can therefore continue with our PCA. [2]

2.2 Principal Component Analysis (PCA)

Principal components are linear transformations of our original variables, whereby the first principal component represents a straight line that best expands the data out. The second principal component has a lesser variance than the first and is orthogonal to the first principal component in such a way that a new coordinate system is made, whereby a set of 21 correlated variables will be transformed to 21 uncorrelated principal components. Although only the first few principal components will be sufficient to represent our data.

The coefficients of these linear combinations of our original variables are called loadings. Variables with small loadings can be ignored. As a rule of thumb, throughout this course we have selected the principal components which cumulatively have accounted for 80% of the variation within our data, but when the sample size is large (as in our case) smaller values can be appropriate. [2]

As we have already covered in our masters program, PCA is used primarily to reduce the dimensionality of multivariate data to simplify our analysis, but how can we represent individual rows of our data? We use scores, which is simply just plugging in our data for the particular row into any of our finalized principal component, and it will give its score for it. By plotting the scores of each of our data point and loadings of our principal components on a biplot, we can identify subgroups of our data points that are similar in nature and recognize what variables contrast these subgroups. The principal components which account for maximum variation in our data will also be later used to plot the clusters, if any, of our data in two-dimensional form.

As our data has variables that are measured on different scales we will be using the correlation matrix to calculate our principal components. We will not go into much detail about the mathematics behind PCA in this report, but rather just apply it to our given data and discuss its results.

2.2.1 Selecting Principal Components

Among the ways to short-list our principal components, we will be mainly using the elbow method and the 80% rule of thumb [3]. The elbow method involves using a scree plot of the variances and identifying the point where it levels off. The 80% rule is choosing enough principal components that account for for 80% of the variation within our data. Although we can reduce this amount for large sample size as in our case. The following table shows the result of our PCA featuring first 8 principal components:

Table 2.1: A summary of our first 8 Principal Components

Principal Component	Standard Deviation	Proportion of Variance	Cumulative Variance
PC1	2.72	35.41%	35.41%
PC2	1.48	10.55%	45.96%
PC3	1.25	7.5%	53.47%
PC4	1.17	6.6%	60.05%
PC5	1.12	6%	66.08%
PC6	1.11	5.9%	72.00%
PC7	1.03	5.1%	77.11%
PC8	0.93	4.1%	81.242%

This shows that the first 8 principal components account for 80% of the variation within our data, but after the first 3 principal components, individual contribution to variance is not only very low but also drop in variance is less. The following figure shows the scree plot of the variances:

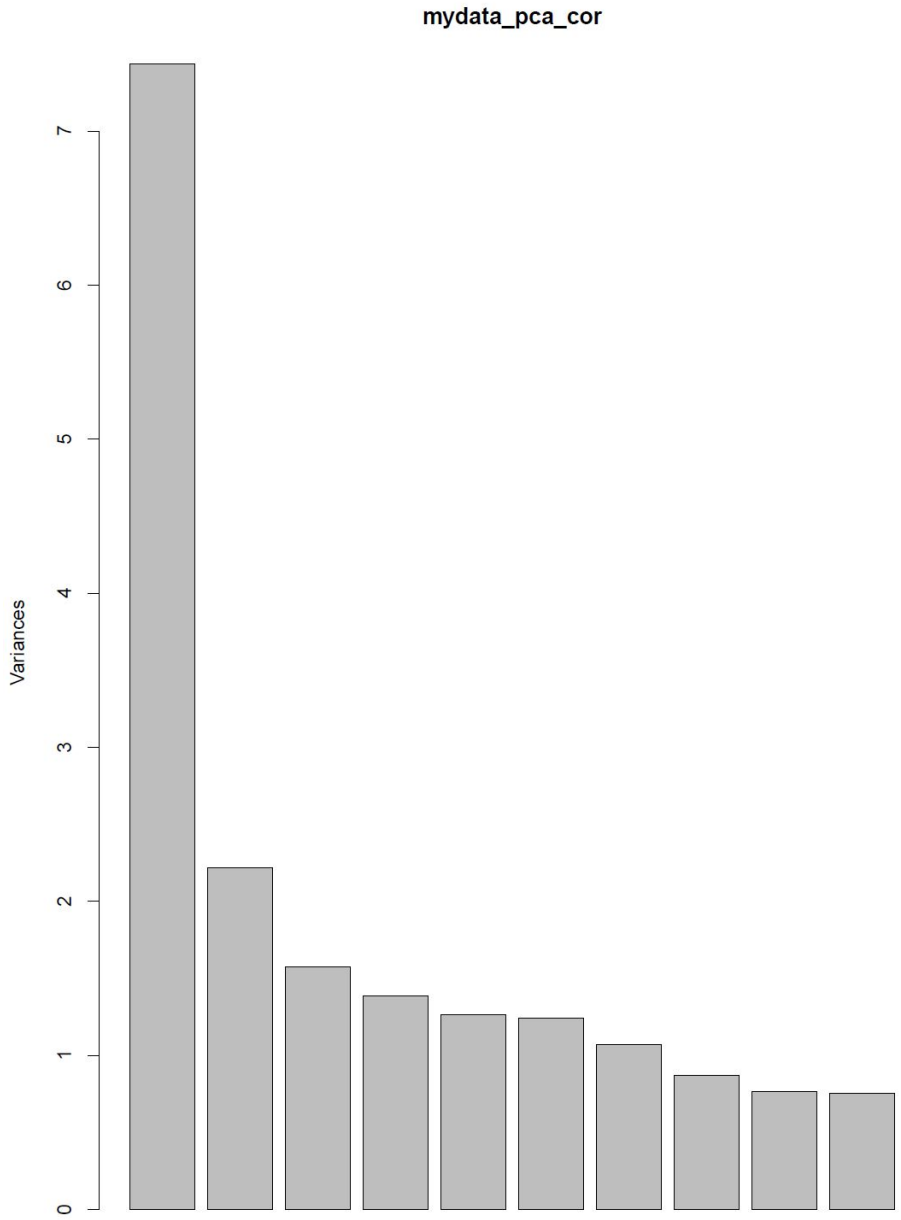

Figure 2.2: Scree plot of variances for our principal components

The scree plot shows that the levelling off occurs after the first four principal components. The first four account for almost 60% of the variation within our data. Is this justified? We have to see the biplots if it is. If we can find no pattern within a principal component with low share of individual contribution to variation (like the 4th principal component and beyond it), then we can neglect it and those after it and use the preceding principal

components to carry out our analysis.

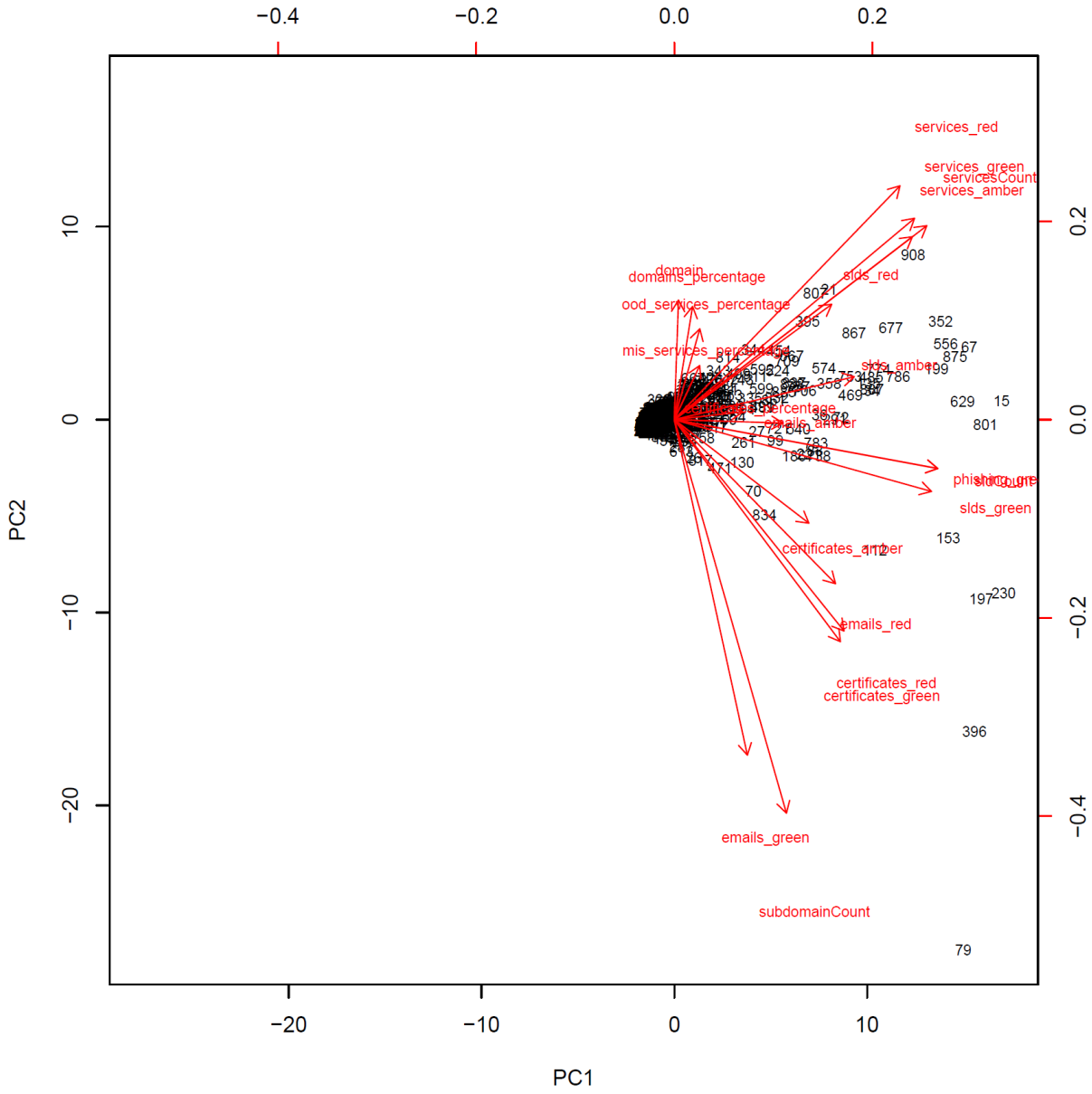

Figure 2.3: Biplot of PC1 vs PC2

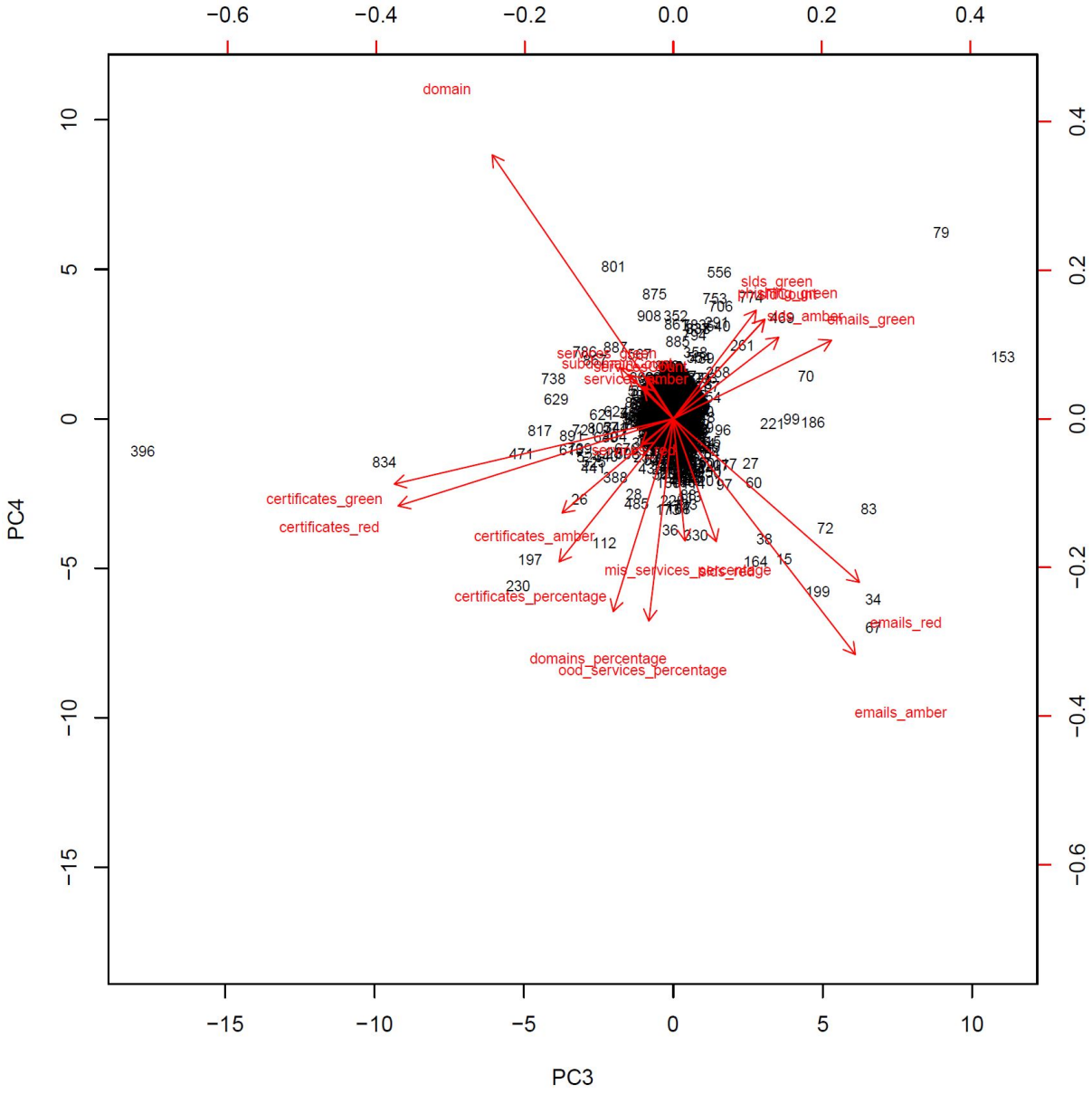

Figure 2.4: Biplot of PC3 vs PC4

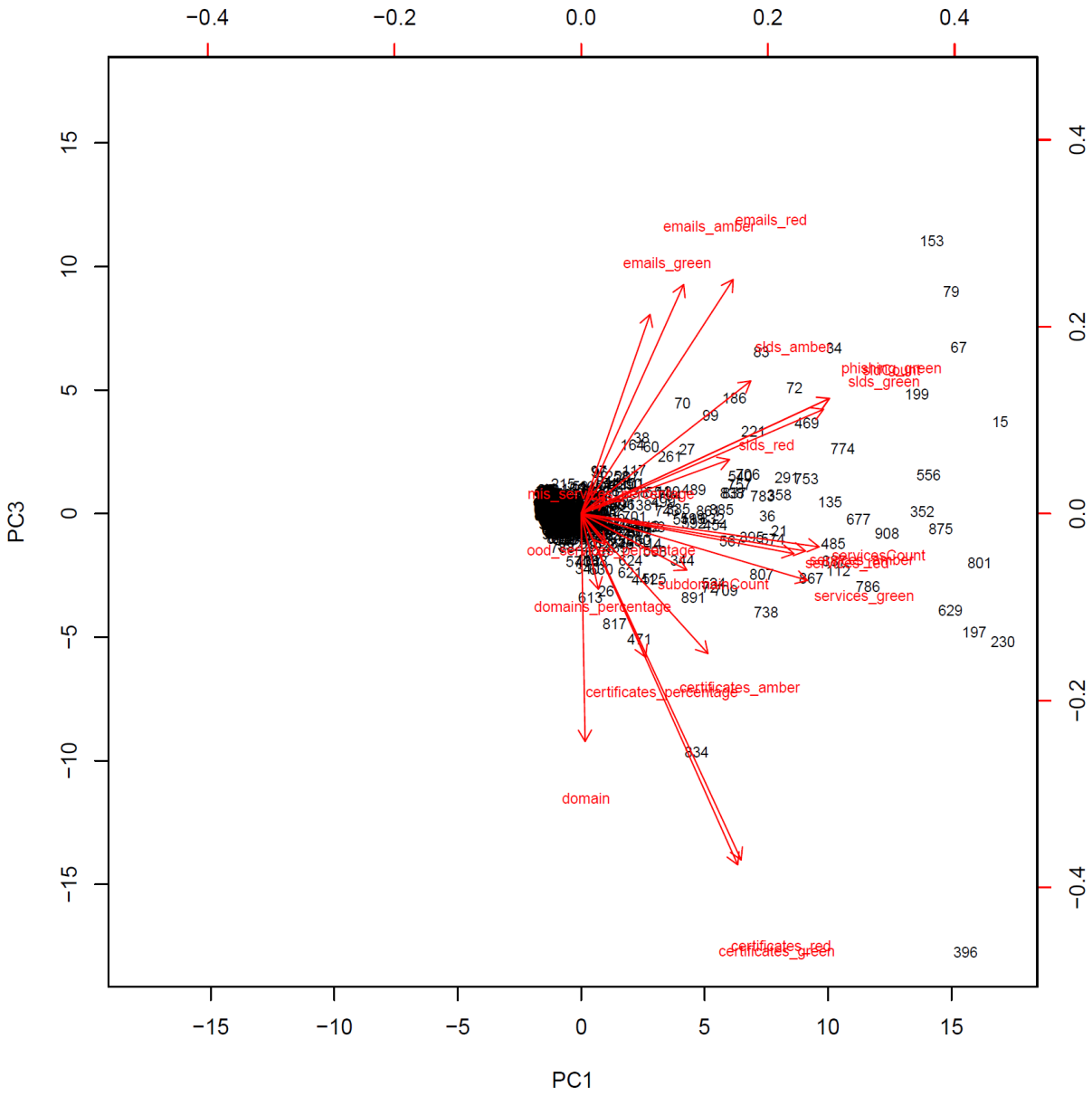

Figure 2.5: Biplot of PC1 vs PC3

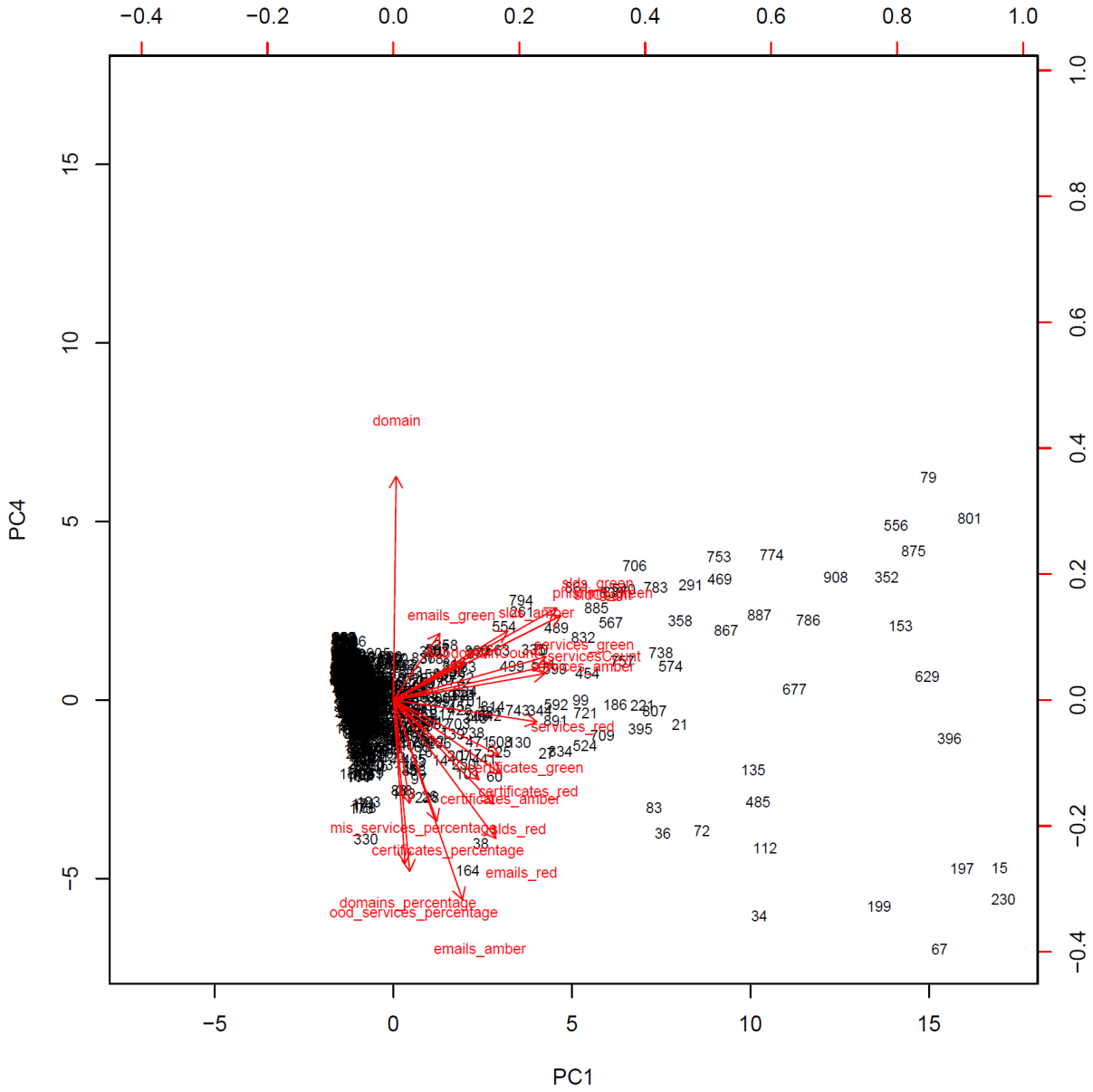

Figure 2.6: Biplot of PC1 vs PC4

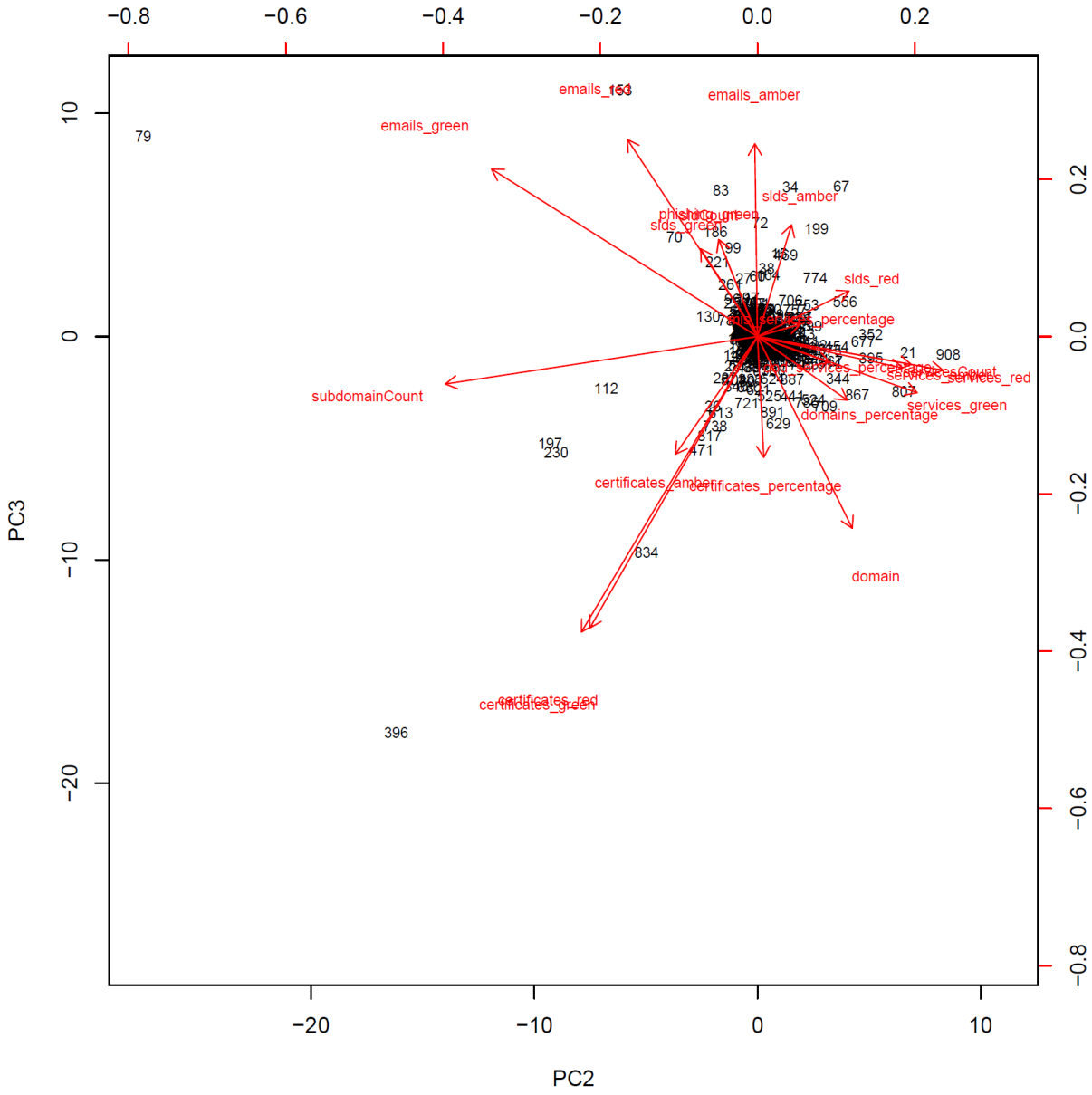

Figure 2.7: Biplot of PC2 vs PC3

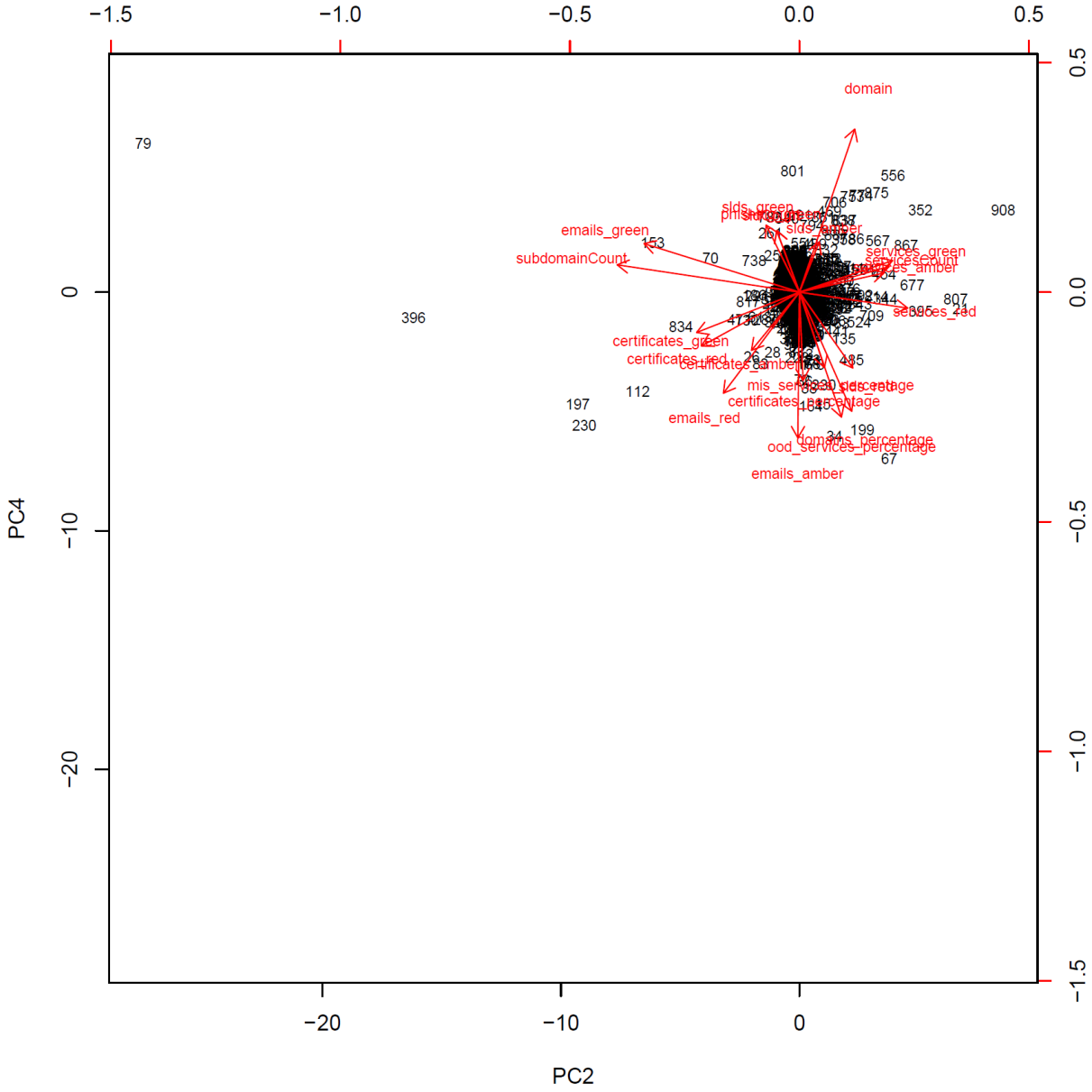

Figure 2.8: Biplot of PC2 vs PC4

2.2.2 Conclusions from Biplots of PCA

The *first principal component* is just an average of all the metrics of a company's website. So, company 15 and 230 make up the most number of everything. Most prominent features with high loadings are certificates and services risks (of all colors), red risks for e-mail, green risk for phishing, domain risks (of all colors), and domain count. Most companies are centred around the origin with low scores. We ignore metrics with small loadings.

The *second principal component* shows an interesting contrast between the **services** and

e-mail + certificates metric. Which means that if any one of them is higher, the other metric will be lower i.e. if a company has a higher number of inbuilt services on its website which are in "red" (i.e at risk), amber (medium risk) and green (low risk), then its risks for company e-mail and security certificates (more specifically red and green, as loading for amber is small) are lower and vice versa.

It can also be seen that when more services are at risk (red), then percentage of out of date services and mis-configured services are also higher, which makes sense. Also, when the subdomain count is higher (it has high loading), the percentage of domains which are at risk is lower (domains_percentage) and the red and amber risks for domain are lower.

The *third principal component* is a contrast between the **e-mail**s and the **certificates** i.e. when one metric is lower, the other is higher. Small loadings are neglected here. It can also be seen that when risks for certificates are higher, percentage of expired certificates is also high which is logical.

The *fourth principal component* and others after it have very low share of variance so it is not beneficial to study them since they do not cater to the a major share of variation in the whole data. [4]

We have therefore seen that some contrasts in our PCA have clustered some companies, so we expect to see these clusters in one of our clustering algorithms.

2.3 K-means Clustering

This is one of the clustering techniques we will apply to our data. K-means clustering clusters data into "K" clusters. We will not go into much detail about the mathematics behind it, but we will delineate in brief how the algorithm works.

We start by randomly fixing an observation to a specific cluster from 1 to K. This serves as our initial cluster. We then find the centroid of each cluster and assign observations to those clusters which are closest to their centroid. By closest we mean the shortest euclidean distance to the centroid in our case. This is iterated multiple times until cluster allotment stops changing. Before carrying out this clustering technique, data must be scaled so it is comparable since the frame of measurement for some features is different than others (like percentage). This means that all our features within the data will be standardized to have a standard deviation of 1 and mean of 0.

The challenge we have at our disposal now is finding out the value of K. This can be done through three methods. [5]

2.3.1 Finding the Value of K

1. **Elbow Method**

   One of the main aims of our K-means clustering technique is to minimize the intra-cluster variation. The elbow method plots the within-cluster sum of squares, which should optimally be as small as possible, for each value of K. The location of a knee i.e. the point where the graph levels off is considered the optimal value of K.

   For our data, we get the following figure:

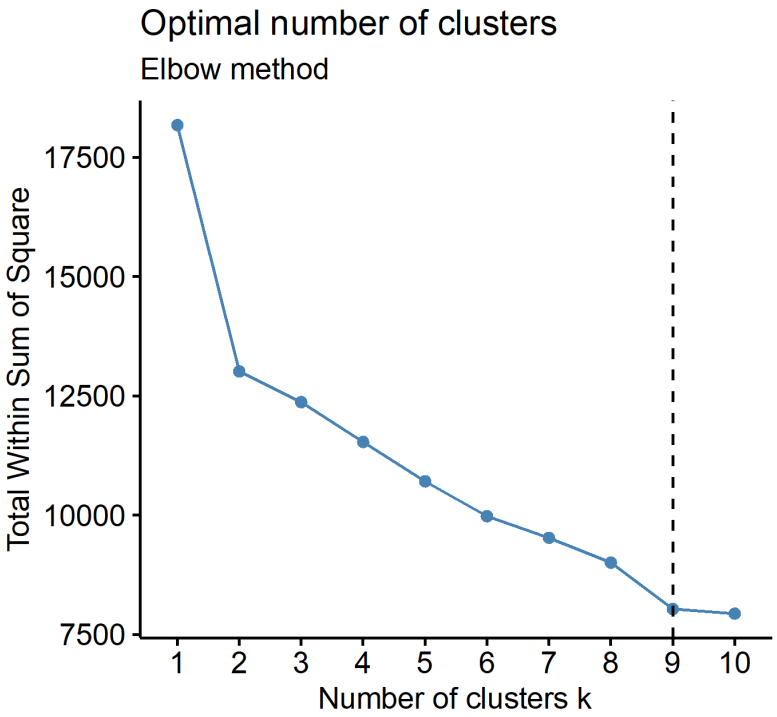

Figure 2.9: Elbow Method to decide K

   Graph seems to level off after 9 clusters.

2. **Silhoutte Method**

   This method measures the average silhouette width for each value of K, a higher value of which indicates good quality of clustering i.e. how well a data point fits within a cluster. We get the following result for our data:

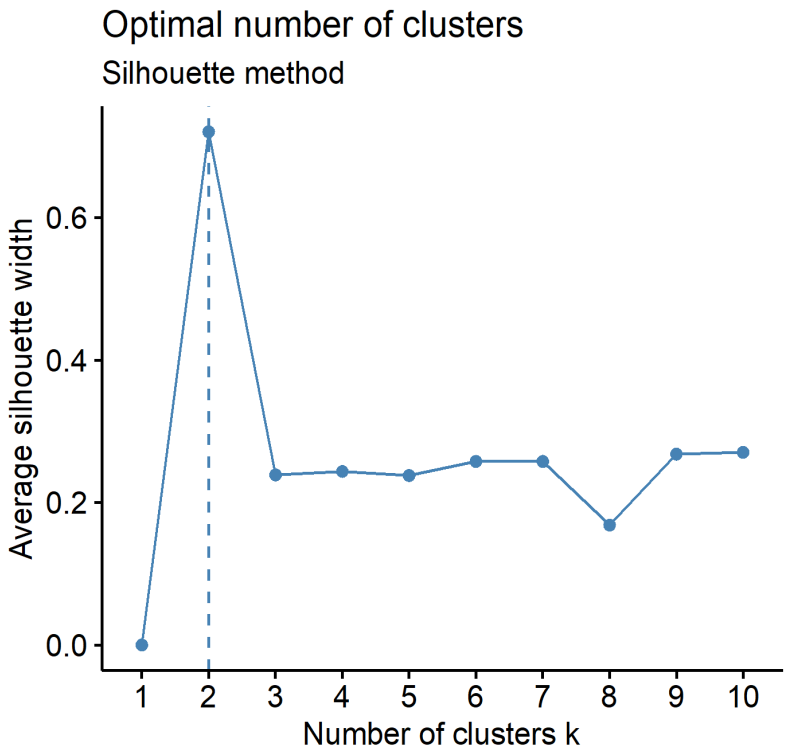

Figure 2.10: Value of K with the Silhoutte Method

3. **Gap Statistic**

   This Method compares the intracluster variation starting from 1 to K with what is expected under the reference null distribution. The full theory behind this method can be found here: [6]. When applied to our data in R studio, we get the following result:

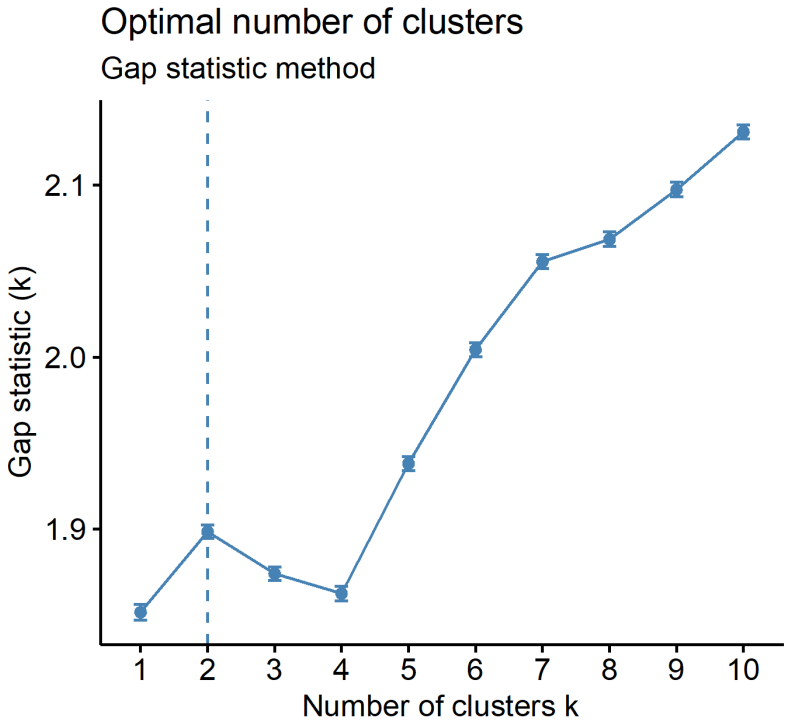

Figure 2.11: Gap Statistic method to find K

The highest value of K for the gap statistic before it goes down is 2. So K =2 is finalized with this method.

Conclusion

We will use both K=9 and K=2 and make sense of our results with our PCA.

2.3.2 K-Means Clustering Result

With K = 9 and K= 2 we get the following clusters plotted on our first two principal components which represent the majority of the variance within our data.

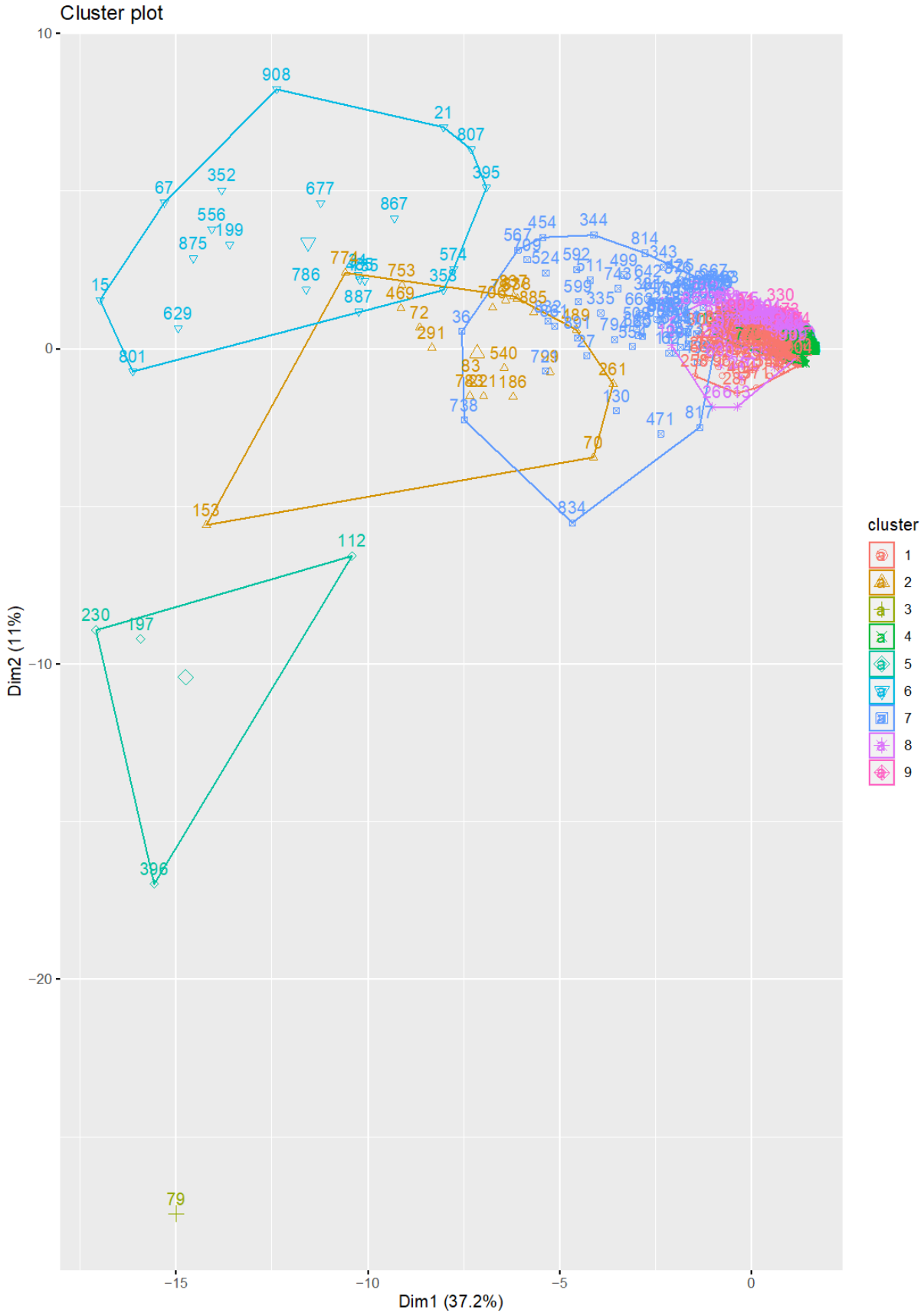

Figure 2.12: Clustering with K=9 on first two principal components

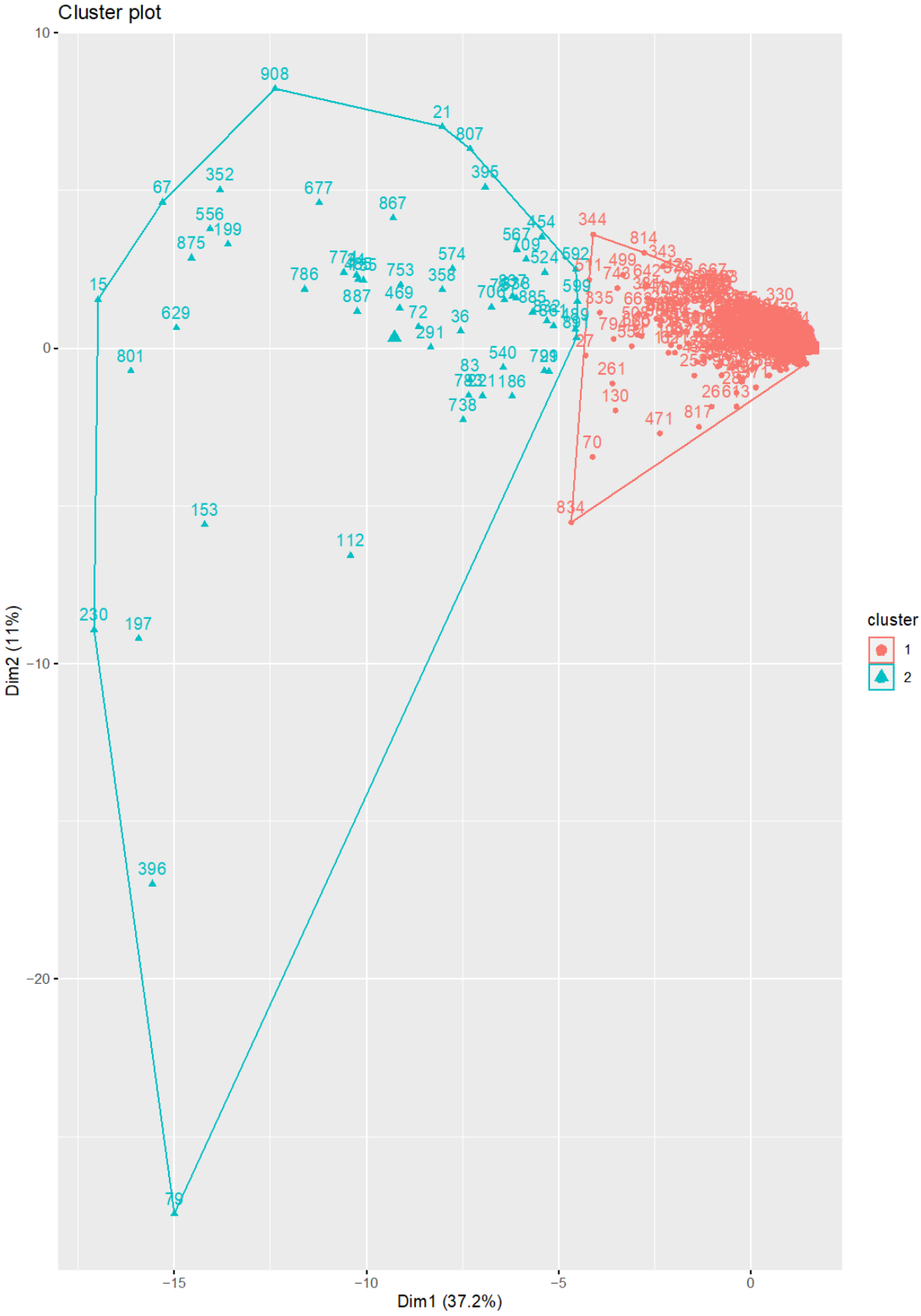

Figure 2.13: Clustering with K=2 on first two principal components

2.3.2.1 Conclusions from K-means Clustering

When $\mathbf{K = 9}$, we get very interesting results with fascinating contrasts. Cluster 5 and 6 are representing our *second principal component* conclusion. Cluster 5 has companies which have higher certificates along with email metrics but low service metrics. Cluster 6 has the opposite.

Cluster 2 and Cluster 7 represent our *third principal component*. Cluster 2 has higher e-mail metric and lower certificate metric with some overlap with cluster 6, while Cluster 7 behaves the opposite with some overlap with cluster 2. Data point of 79 is an outlier so should be neglected.

All other clusters have high overlaps, which makes sense as other principal components didn't account for much variation in the data. Also, the cluster with highest number of points is cluster 1. Most data points in our principal components were centred around the origin, the majority of which then end up in our cluster 1. A closer inspection of these clusters is as follows:

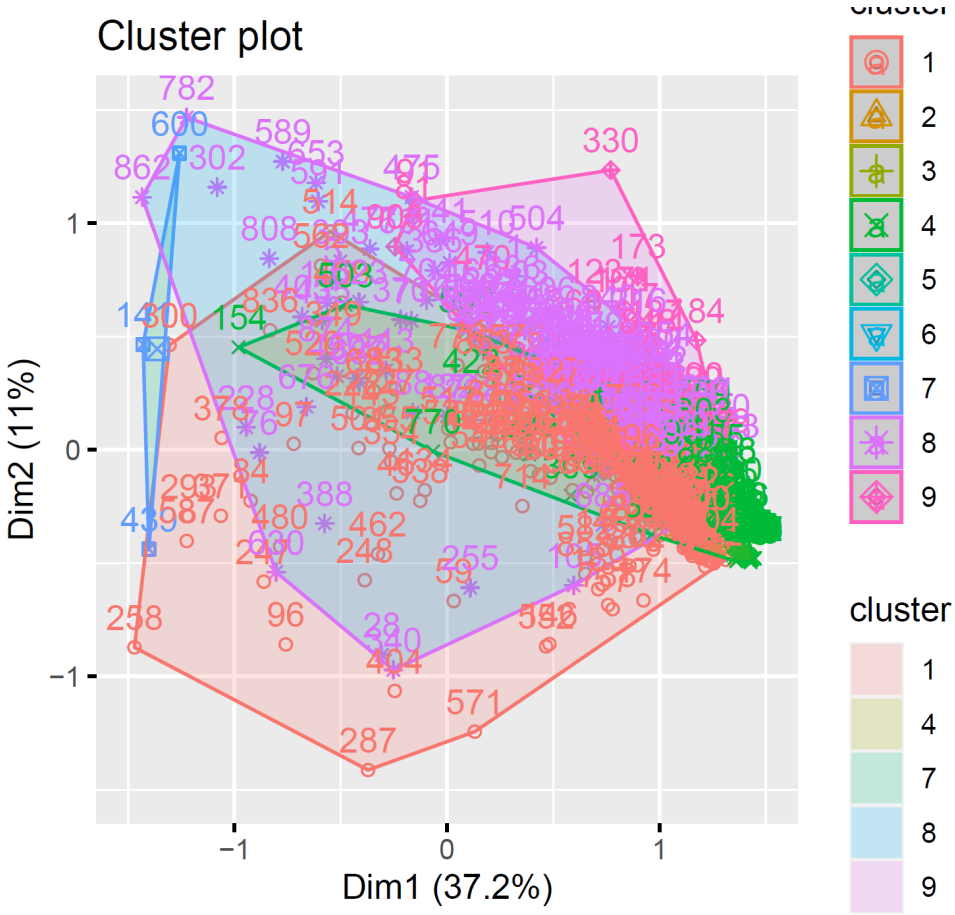

Figure 2.14: Zooming into overlapped clusters when K = 9

It can be seen that cluster 1, 4 and 8 have highest number of members with a lot of overlapping. The reasons for these forming is unknown and our principal components including and after the fourth one do not give an adequate explanation for these forming since they make a small part of our total variation in the data.

When $\mathbf{K = 2}$, we get more defined clusters with no overlap, which can be explained by our *first principal component*. Cluster 1 has companies with low weighted average of all cyber risk metrics, while cluster 2 represents the opposite.

2.4 Hierarchical Clustering

Finally we need to see whether hierarchical clustering technique can pickup the previous clusters we have made. In this method we do not need to define the number of clusters we want, rather the algorithm groups them by itself. We will not go into much detail about the mathematics behind this technique but will just delineate how the algorithm works for our case. The algorithm works iteratively by using a dissimilarity matrix formed by measuring the euclidean distance between each pair of our data points and then clustering those points which are most similar to each other. Initially each data point is treated as a cluster and the algorithm proceeds to form clusters until we get one big cluster. This is all represented on a **dendogram** [7]. A dissimilarity between two groups of observations is decided by the **linkage**. In our course we encountered **single linkage** and **complete linkage** and we will use both for our data in our analysis.

2.4.1 Dendograms

First we scale the data then plot the dendogram with complete linkage first:

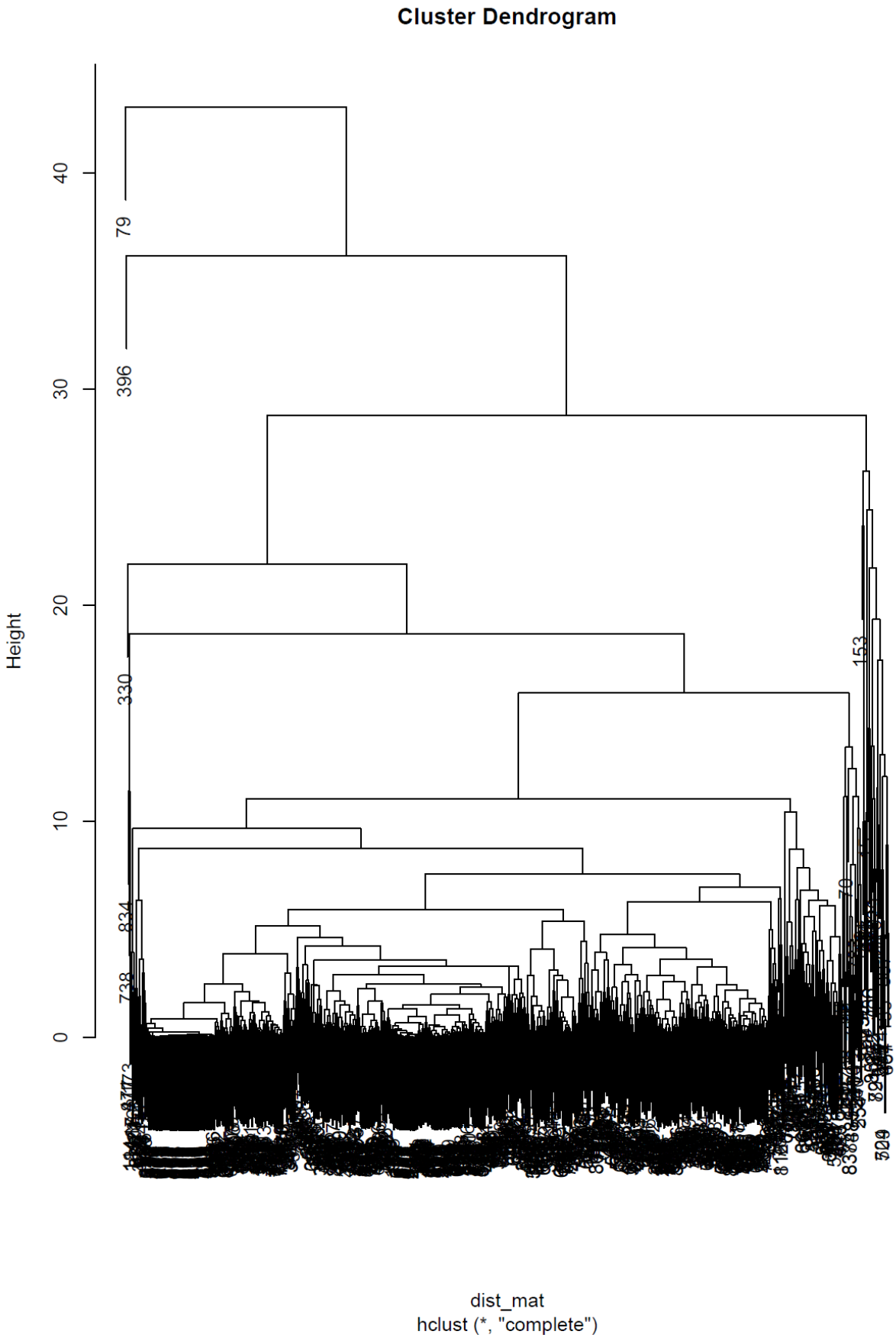

Figure 2.15: Dendogram with complete linkage

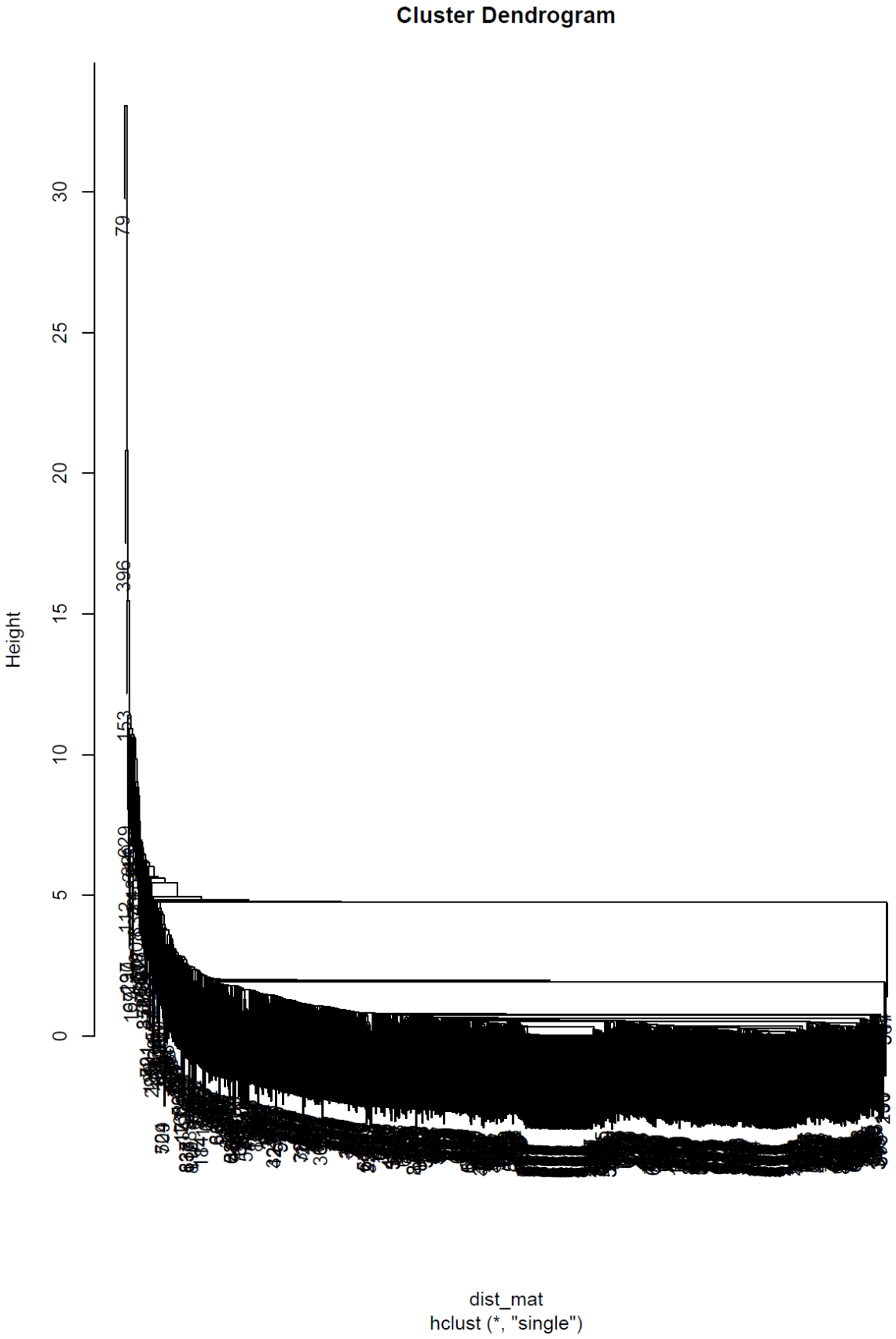

Figure 2.16: Dendogram with single linkage

2.4.2 Conclusions from Hierarchical Clustering

Single linkage clustering did not provide us with a clear result. On the other hand complete linkage provides us with some insight that there are actually a lot of clusters that can be formed from the data. If we cut the dendogram to have 9 clusters only as in our K-means clustering example we get the following result:

```
> cutree(hclust_comp, k = 9)
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1
 [45] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1
 [89] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[133] 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[177] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[221] 1 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[265] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[309] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 8 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1
[353] 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 9
[397] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[441] 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[485] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1
[529] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 1 1 1 1
[573] 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[617] 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[661] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[705] 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[749] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
[793] 1 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[837] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 3 1 1 1 1 1
[881] 1 1 1 1 1 1 3 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1
```

Figure 2.17: Cutting dendogram for complete linkage to have 9 clusters

This shows most observations will then fall into a single cluster i.e. cluster 1. Maybe there is more to the data that we are missing? We expected more spread out clusters like in K-means clustering, but sadly this analysis technique falls short of providing us with any valuable insight. We will evaluate this later on in our discussion.

Discussion

3.1 Conclusions from our Data Analysis

It seems like PCA and K-means clustering have helped in achieving our two main objectives. We have contrasted companies mainly through our principal components and have found out what metrics lead to more red, amber and green risks of e-mails, certificates and services. Other metrics did not make much of a cut due to their low loadings or they featured in principal components with less share of total variation in our data. We then saw these companies subgroup themselves through our clusters in K-means clustering, and simultaneously verified our PCA. Hierarchical clustering on the other hand was not much help to us. This is what our data analysis tells us so far, but we need to evaluate these results before taking them as a fact.

It is to be noted that since our data was scaled before carrying out our analysis, the concept of "high" and "low" of any metric is relative to the variation of that metric in our data. For example a "high" number of red certificates may constitute a number of 10,000 because it is close to our maximum value for this feature, but the same cannot be said about another feature. Therefore when we say high e-mail + certificate metrics lead to low services metrics and vice versa, what we can expect is that in the actual data (not scaled) the values for these may not look like "high" and "low", but given the context of the whole data of these features when they are scaled, they are actually assigned high and low status.

3.2 Solution Evaluation

Careful consideration needs to be applied when taking our conclusions from our data analysis as purely legitimate. We could have reached a different conclusion if more data had been at our disposal. Similarly conclusions from unsupervised learning are hard to assess as there is no validation mechanism in place as it is in supervised learning where the data is already labelled. Therefore we will never know if our conclusions hold weight as we cannot test our model and have to make sense of trends in the data on our own.

In our PCA of the data, using the third principal component to make conclusions was

quite generous as it explained less than ten percent of the variation in our data, and also provided a contradicting claim that certificate metrics rise when e-mail metrics fall and vice versa, but the second principal component claimed both these metrics to rise together when services metrics fall and vice versa. Later principal components did not also provide us with a relevant trend. Therefore conclusions reached from the first two principal components should be given more legitimacy than those after it since they both have greater share of variance for our data. Even though our third principal component did provide us explanation for two clusters formed in our K-means clustering, but those clusters had a lot of overlapping with other clusters. In conclusion, there is no authorized way found in literature to decide on how many principal components are enough. At the end of the day this approach will remain subjective in nature.

In our K-means clustering method, we had difficulty in deciding on the value of K, i.e. the number of clusters. There are other methods not explored in deciding the number of clusters like the *hypersphere density based approach* [8]. The value of K greatly shifts our conclusions on the data so it is vital to choose a pertinent value for it. It should also be noted that this technique is highly sensitive to outliers and can give different results if the layout of the data is changed. We can therefore use other techniques like *Partitioning Around Medoids (PAM) clustering* [9] to tackle this issue. We also have no guarantee that if we receive a new set of data, will they form the same clusters for the same value of K? We thus cannot validate our conclusions from our data analysis. One approach could have been to cluster subgroups of the data and see if they behave in a similar manner, but such practices would require more time.

In hierarchical clustering we also have not tested out using other linkages like average and centroid linkage, which could have given us different results. Centroid linkage measures the dissimilarity between the centroids of two clusters and average linkage measures the average dissimilarity between two clusters. There is also the issue of cutting the dendogram to get optimum clusters that properly define our data. In our data analysis, hierarchical clustering deemed unfit in giving us any logical conclusion and instead clustered most of our data points to a single cluster even with K=9.

In both our clustering techniques we have used squared euclidean distance to measure distance between two data points or clusters or between a data point and a cluster. We used it as a dissimilarity measure in hierarchical clustering and to measure variation between clusters in K-means clustering. In hierarchical clustering, other dissimilarity measures can also be used such as generalized squared euclidean distance, the Mahalanobis distance or the correlation-based distance [1] (which might have fared better instead of squared euclidean distance since some of our features are actually highly correlated).

Overall we can say that much more could have been done with the data. More unsupervised learning techniques could have been applied which are outside the scope of what we learned during our degree program or more data could have been analysed or companies could have been inspected in each clusters and specific characteristics about them could have been examined. Data other than cyber security metric could have been inspected for subgroups of companies in our clusters or credit scoring data, if provided, could have been compared with our present data analysis for interesting insights. The possibilities for more insights are definitely present if more time was afforded to us.

Bibliography

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer, 2017.

[2] B. Everitt and T. Hothorn, "An introduction to applied multivariate analysis with r," p. 71–72, 2011.

[3] I. Jolliffe, "Principal component analysis," p. 111–115, 2002.

[4] L. Hayden, "Principal component analysis in r," *Datacamp*, 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/pca-analysis-r.

[5] B. Boehmke, "K-means cluster analysis," *University of Cincinnati*. [Online]. Available: https://uc-r.github.io/kmeans_clustering

[6] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[7] A. Kassambara, "Practical guide to cluster analysis in r," *STHDA*, p. 67–78, 2017.

[8] S. Nanjundan, S. Sankaran, C. Arjun, and G. Anand, "Identifying the number of clusters for k-means: A hypersphere density based approach," in *International Conference on Computers, Communication and Signal Processing*, 2019.

[9] L. Kaufman and P. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," p. 68–126, 2009.

Appendix

R CODE USED FOR DATA ANALYSIS:

IMPORT AND PREPARATION

```
mydata = read.csv("project_data.csv",header = TRUE) #importing data
attach(mydata)
mydata$X.orgId <- NULL #removing all unnecessary data
mydata$orgName <- NULL
mydata$userId <- NULL
mydata$domain <- 1:nrow(mydata) #replacing domain column with row number
mydata$domain <- as.character(as.numeric(mydata$domain)) #treat domain column
as character
mydata$mis_services_classification<- NULL #removing all factor data
mydata$domains_classification<- NULL
mydata$ood_services_classification<- NULL
mydata$certificates_classification.<- NULL
mydata$domains_numerator<- NULL # removing all fraction data
mydata$domains_denominator<- NULL
mydata$certificates_numerator<- NULL
mydata$certificates_denominator<- NULL
mydata$mis_services_numerator<- NULL
mydata$mis_services_denominator<- NULL
mydata$ood_services_numerator<- NULL
mydata$ood_services_denominator<- NULL
summary(mydata) # all columns with constant values to be removed for scaling
mydata$credentials_red<- NULL
mydata$credentials_amber<- NULL
mydata$credentials_green<- NULL
mydata$phishing_red<- NULL
mydata$phishing_amber<- NULL
mydata$ransomware_red<- NULL
mydata$ransomware_amber<- NULL
```

```
mydata$ransomware_green<- NULL
library(dplyr)
df <- mydata %>%
  mutate_at(c(2:21), funs(c(scale(.)))) #scaling the data
```

PRINCIPAL COMPONENT ANALYSIS

```
library(corrplot)
corrplot(cor(mydata), method="ellipse") #correlation plot to check
if correlation matrix needed
#domain number would be ignored in PCA
mydata_pca_cor <- prcomp(apply(df[2:21],2, scale))
mydata_pca_cor #specifications of PCA
screeplot(mydata_pca_cor) #plots the scree plot
summary(mydata_pca_cor) #summarize the result of PCA and give
variation percentage
library(ggbiplot) #now biplots constructed of first 4 principal components
ggbiplot(mydata_pca_cor)
biplot(mydata_pca_cor, choices=c(1,2), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
biplot(mydata_pca_cor, choices=c(3,4), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
biplot(mydata_pca_cor, choices=c(1,3), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
biplot(mydata_pca_cor, choices=c(1,4), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
biplot(mydata_pca_cor, choices=c(2,3), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
```

```
biplot(mydata_pca_cor, choices=c(2,4), cex=0.7,col=c("black","red"),
scale = 0, xlabs
=
     row.names(mydata))
```

K-MEANS CLUSTERING

```
library(factoextra)
library(NbClust)
#Elbow Method
fviz_nbclust(df[2:21], kmeans, method = "wss") +
  geom_vline(xintercept = 9, linetype = 2)+
  labs(subtitle = "Elbow method")

# Silhouette method
fviz_nbclust(df[2:21], kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")

# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
set.seed(123)
fviz_nbclust(df[2:21], kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")

# Compute k-means with k = 9 for elbow method
set.seed(123)
km.res <- kmeans(df[2:21], 9, nstart = 25)

# Print the results
print(km.res)

#paste results of clusters to the main data
dd <- cbind(mydata, cluster = km.res$cluster)
head(dd)

#plot clusters on first two Principal Components
fviz_cluster(km.res, data = df[2:21])
```

```
#zoom in for a better view of overlapping clusters
fviz_cluster(km.res, data = df[2:21]) + xlim(-1.5,1.5) + ylim(-1.5,1.5)

# Compute k-means with k = 2 for other methods
set.seed(123)
km.res <- kmeans(df[2:21], 2, nstart = 25)

# Print the results
print(km.res)

#paste results of clusters to the main data
dd <- cbind(mydata, cluster = km.res$cluster)
head(dd)

#plot clusters on first two Principal Components
fviz_cluster(km.res, data = df[2:21])
```

HIERARCHIAL CLUSTERING

```
dist_mat <- dist(df[2:21], method = 'euclidean') #dissimilarity matrix
formed with squared euclidean distance
hclust_comp <- hclust(dist_mat, method = 'complete') #complete linkage
used
plot(hclust_comp) #dendogram plotted
cutree(hclust_comp, k = 9) #tree cut to have 9 clusters and data point
pasted with their cluster
dist_mat <- dist(df[2:21], method = 'euclidean')
hclust_comp <- hclust(dist_mat, method = 'single') #single linkage used
plot(hclust_comp)
cutree(hclust_comp, k = 9)

#this is extra step to plot heat map of clusters. Can be used for visually
seeing changing clusters at the roots
suppressPackageStartupMessages(library(dendextend))
avg_dend_obj <- as.dendrogram(hclust_comp)
avg_col_dend <- color_branches(avg_dend_obj, h = 2)
plot(avg_col_dend)
```