Statistical Data Analysis
Project 1

Jef Masereel, r0631651, group 19

April 23, 2020

Exploratory analysis & transformation to normality

1.1 Exploratory analysis

In order to do a first inspection of the given dataset, a few tools are available. The first step is a visual inspection of the variables and their summary (in R). This already highlights a few differences. Although most variables are continuous measurements, three are not. The variables 'TypeOfSteel_A300', 'TypeOfSteel_A400' and 'Outside_Global_Index' are binary values (0/1). They are removed from the dataset for all further steps. Another observation from the first inspection is that 'Orientation_Index' and 'Luminosity_Index' have values in [-1:1] whereas all other continuous variables are positive. Note that some are not strictly positive, as discussed further in 1.2. Lastly, the 'Flt' variable seems to be a classification label. All observations fall under three possible values: 'Z_Scratch', 'K_Scratch' and 'Other_Faults'.

By computing the boxplots, it is clear that many of the continuous variables are far from the perfect normal distribution. Some contain many observations that are considered outliers by the boxplot method (normality assumption). Most are skewed to one side, although in varying degrees. Since the next subsection discusses normality in more detail, figures of the boxplots are excluded to leave room for the next sections. They can be generated by the R code included with this report if needed. A pairwise clusterplot (generated by code, left out due to report size requirements) of all continuous variables visually demonstrates some interesting distributions and interdependencies. Some clusterplots show very dense concentrations of values around a specific center, while others show a broader distribution. In the case of 'Length_of_Conveyer', there even seem to exist two separate clusters. Another interesting observation is the presence of linear and logarithmic relations between for example 'X_Minimum' and 'X_Maximum'. All of this together suggests it might be worth looking into underlying distributions, as will be elaborated in the next sections.

1.2 Transformation to normality

In the code enclosed to this report all continuous variables are thoroughly inspected by computing their histogram, density plot, QQ-plot and Shapiro-Wilkes score. The last scores quickly indicate the degree to which each variable matches the univariate normality assumption. In this case, this assumption is strongly invalidated for all, with the lowest P value going down to 6.8e-48. To demonstrate how one could attempt to transform a univariate dataset towards normality, the enclosed code applies the BoxCox transform to 'Empty_Index', which has the highest observed P value (2.147074e-05) of the 22 continuous variables. Figures 1 and 2 show the limited effect of this transformation, meaning the assumption of univariate normality is still not validated. The Shapiro Wilkes score confirms this, after the transform it has not improved (3.890283e-05). Note that one outlier of value zero had to be removed (temporarily) to satisfy the requirement of strictly positive data for the BoxCox transform. All other figures can be generated by the enclosed code, but would make this report too large. It suffices to note that all other variables have lower P values in the Shapiro Wilkes test, and that the dual peaks in the observations suggest the presence of an underlying multivariate distribution. In the further sections, the variables are used with their original variables.

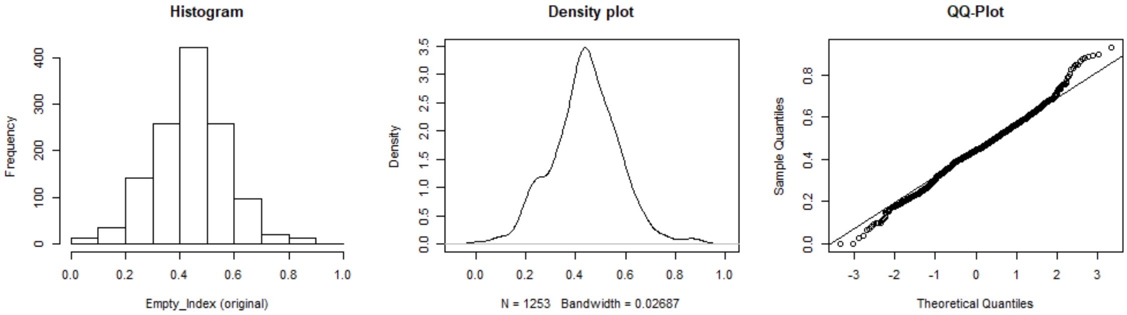

Figure 1: Univariate normality inspection of original Empty_Index

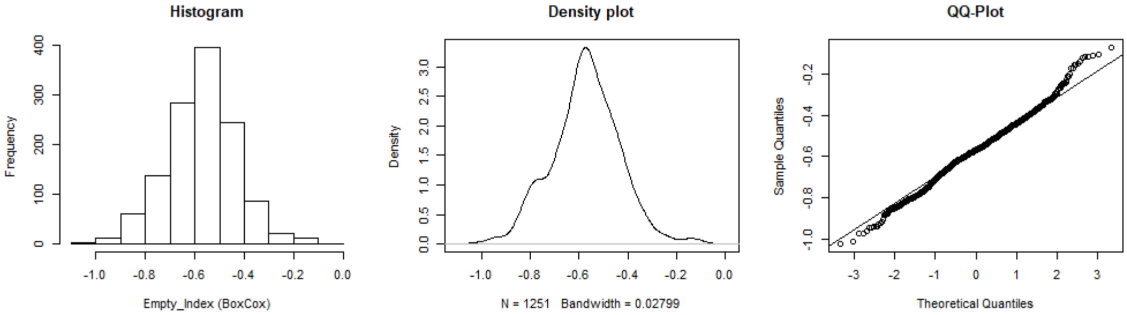

Figure 2: Univariate normality inspection of Empty_Index after applying BoxCox

PCA

2.1 PCA analysis

Before computing the PCA, the dataset is reduced to its 22 continuous variables as described in the previous section. Eventually, no variables underwent a BoxCox transformation since they are all too far off from the univariate normality assumption. Since the variables vary in amplitude of observations and no units are given, it is best to use the scaled PCA. This comes down to computing the eigenvalues of the correlation matrix of the reduced dataset, NOT the covariance matrix. This standardizes the scale of observations across variables. The PCA analysis in the enclosed R code shows that the amount of variability can indeed be condensed into much less variables. The cumulative proportion of information for the first six principal components is 84%, for the first eight this is 91%. Based on the previous section it seems the underlying distribution of the dataset is relatively complex, so it is likely the further steps will need all the information available. This, combined with the fact that no large data compression is required, makes it reasonable to keep the first eight principal components for further computations.

2.2 PC inspection

Figure 3 shows a set of colored, pairwise scatterplots for the first three PCs. K_Scratch (red) has a noticeably different distribution from the other two classes, which suggests that based on these first three components it might be possible to obtain a decently accurate clustering. Despite this first impression, the next section will show that for the considered clustering methods its elongated cluster shape is very hard to model properly. The two other types however seem to have much more overlap, although Other_Faults (green) has a larger variance for the shown components than Z_Scratch (blue).

The diagnostic plots shown in figure 4 all indicate the expected presence of outliers (everything above the indicated distance threshold), for varying numbers of included PCs. Other_Faults (green) seems to contain the highest number of outliers, which makes sense given its shape in the previous figure. The enclosed code removes no outliers until the last section (after multivariate normality test by MCD estimate).

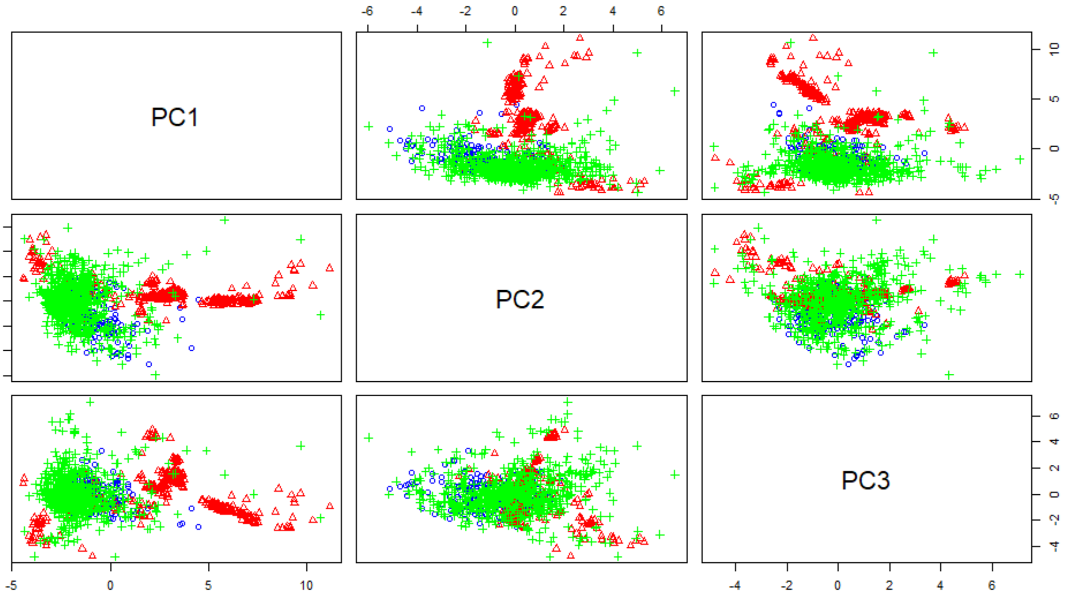

Figure 3: Scatter plots of first 3 PCs. Color legend in text above.

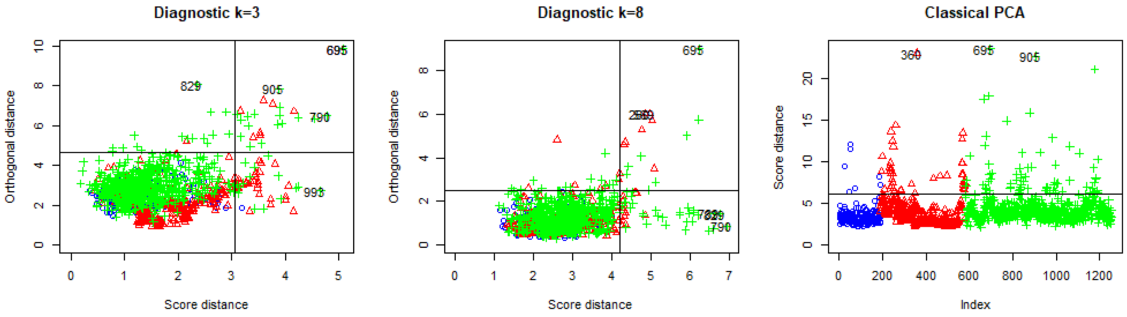

Figure 4: Diagnostic plots for k=6, k=8 and k=22

method	hit rate Z	hit rate K	hit rate O	miss rate	balanced accuracy
PAM	57.65	5.26	81.28	30.81	74.73
Diana	96.32	2.82	80.00	58.98	59.71

Table 1: Comparison of clustering performance

Clustering

3.1 Cluster analysis

After some experimentation (also see 3.3) the best clusterings were obtained with the partitioning around medoids (PAM) method. For hierarchical clustering methods, divisive analysis (diana) performed best. The exact implementation in R is simply:

```
> Dpam <- pam(DPCA8@scores,3)
> Ddiv <- diana(DPCA8@scores,3,diss=FALSE)
```

3.2 Resulting plots

The 1253 observations make it difficult to plot insightful graphs, but figures 5 and 6 are shown as a minimal view on the clustering result. The silhouette plot (for PAM) and banner plot (for diana) are not shown since they are difficult to construct properly for this number of observations. Figure 5 shows the clusplot according to the PAM results. This matches pretty well with the pattern in figure 3. Figure 6 shows the dendrogram generated with the diana method, although this graph alone does not give us much useful information. The metrics (table 1) computed by the enclosed code are much easier to interpret.

3.3 Correspondence between clusters and fault types

While writing the implementation in R, multiple options were tried and tested. The enclosed code includes a small self defined function to compute the hit rates for each cluster, the miss rate and the balanced accuracy [1] by comparing the computed clusters with permutations of the Fault types for each observation. These metrics are shown in table 1 above. Partitioning around medoids (PAM) gives the best results. For hierarchical methods, divisive analysis (diana) returned the highest performance, although lower than PAM. As mentioned in the first section, these clustering methods have difficulty identifying observations of the Other_Faults type. Despite the limited context, it is likely that this type contains many subclasses and should be treated as such if accurate classification is required. The other two fault types are classified surprisingly accurate given the large overlap seen in figure 3.

[1] Carrillo, Henry & Brodersen, Kay H. & Castellanos, Jose. (2014). Probabilistic Performance Evaluation for Multiclass Classification Using the Posterior Balanced Accuracy. Advances in Intelligent Systems and Computing. 252. 347-361. 10.1007/978-3-319-03413-3_25.

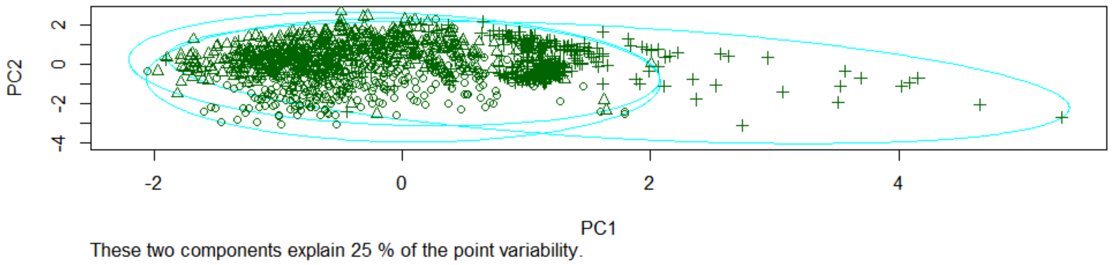

Figure 5: Clusterplot of PAM results. Longest ellipse matches Other_Faults.

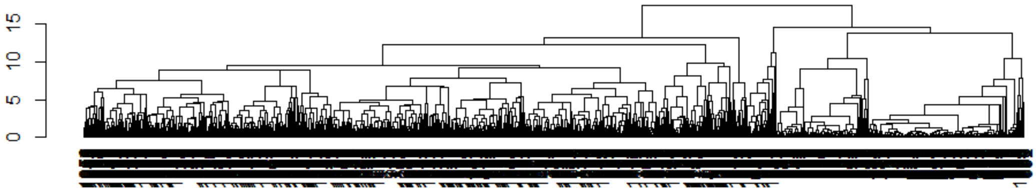

Figure 6: Dendrogram as obtained by diana method. The indices are shuffled...

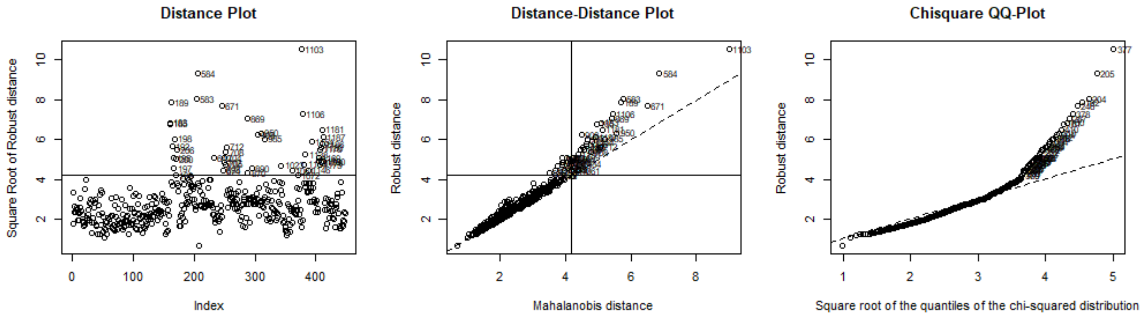

Figure 7: Outlier detection by MCD estimate, applied to scores (8 PCs)

Testing multivariate normality

4.1 Outlier detection

This section focuses on the largest cluster found by PAM, which (based on inspection tool, see code) seems to match quite well with the fault type Z_Scratches. By computing the Minimum Covariance Distance (MCD) estimate for this subset of observation scores (for first 8 principal components), we obtain the plots in figure 7. These clearly indicate the presence of many outliers (93 counted, see code). Especially in the Chi-squared QQ-plot there is a clear run-out of observations in the top-right that deviate from the first bisector. These outliers are removed from the cluster for the next subsection.

4.2 Underlying multivariate normality

Now that the outliers (as indicated by MCD estimate) are removed from the cluster of interest, the underlying distribution can be properly inspected. Figure 8 shows the QQ-plot generated from the PCA scores (8 PCs) for the 402 remaining outlier-free observations in the cluster. The result seems to stay relatively close to the first bisector line. Therefore, I would assume that the underlying distribution of the PCA scores for this cluster is indeed multivariate normal.

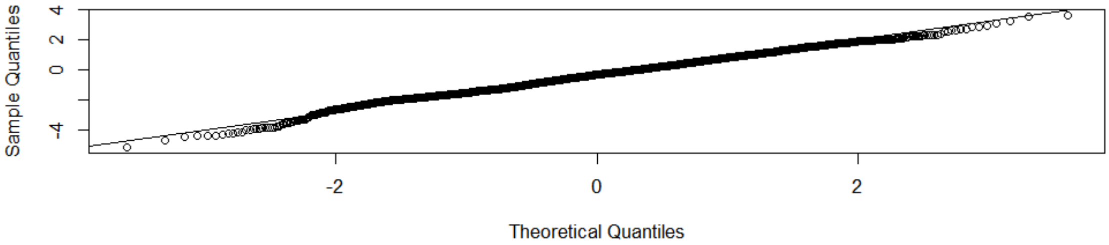

Figure 8: QQ-plot of scores (8 PCs) after outlier removal