PRACTICAL DATA SCIENCE WITH R - MAT00058H - HOMEWORK 2

MATTHEW KNOWLES

1. Corrolations

We begin by looking at the corrolations of the variables in the dataset. We do this using the *cor()* function, and coerce the resulting object into a matrix by using the *as.matrix()* fucntion.

Seen as this matrix is rather large and hard to read a plethera of numbers, we instead chose to use a heatmap to visualise the corrolations better. In the below figures we see the resulting heatmaps.

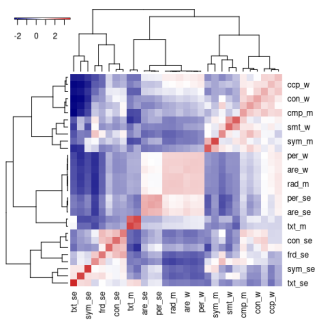

Figure 1. Scaled corrolation heatmap

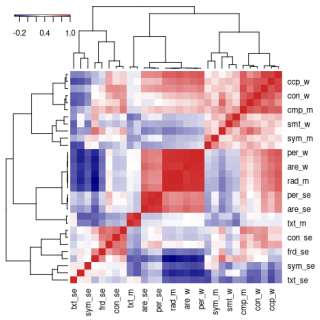

Figure 2. Unscaled corrolation heatmap

We can see a lot of lower corrolations, but there appears to be a lot more positive corrolation on the middle variables. This is more pronounced in the unscaled corrolation, as is to be expected.

2. PCA

We now move on to some PCA. We begin by taking the data as-is to see how it looks. The biplot for this is seen below.

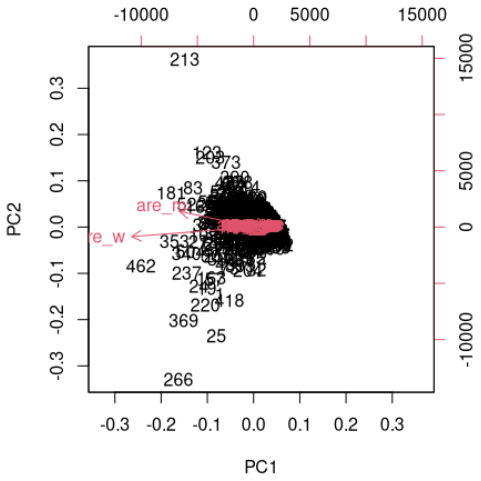

Figure 3. Biplot for intial PCA- no scaling

Date: Spring Term, Week 3.

In this plot we can see that there are 2 data points we may consider to be outliers- 213 and 266. Before deciding whether to remove these and whether or not to scale the data. We produced a boxplot to see which variables are contributing the most to the data.

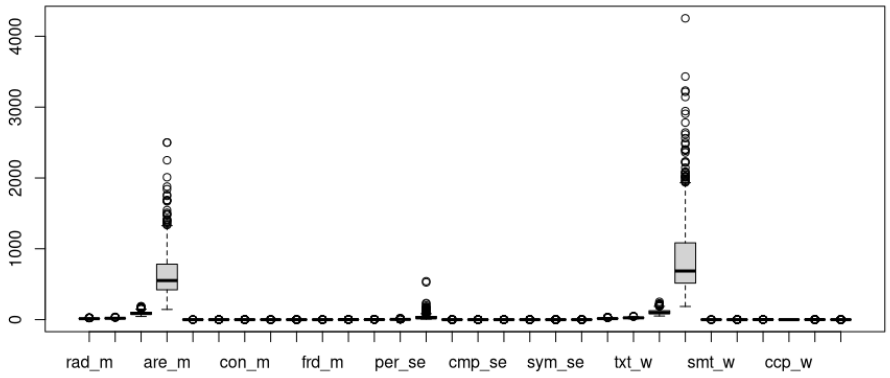

FIGURE 4. Boxplot of variables

This shows clearly that the area variable, both in mean and max value is dominating the analysis, so scaling is a wise idea. This gives an updated biplot as follows:

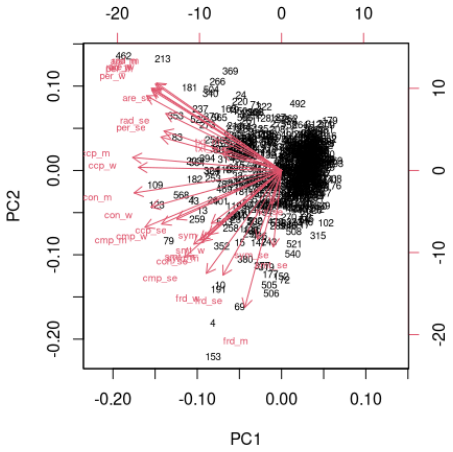

FIGURE 5. Scaled biplot

Now we can see here that there are still some outliers, and 213 is still amongst this crowd. We identify 4, 153, 123 and 462 as outliers, and remove them by creating a new dataset which just leaves them out. This probably isn't the most efficient way to do this, but it works quite nicely. We then finally perform one last PCA to give the following biplot:

With that sorted we can look at the loadings and the corrolations between them. The heatmap below is produced in an almost identical way to the one earlier.

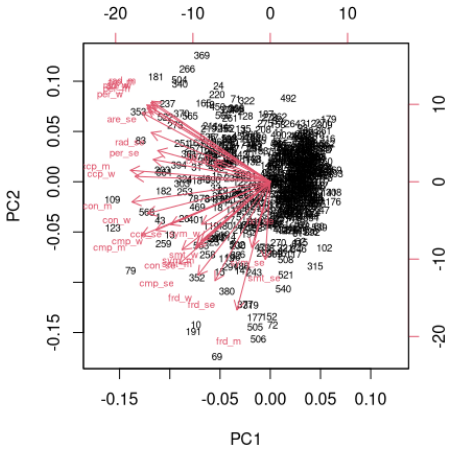

FIGURE 6. Biplot of scaled and outlier-free dataset

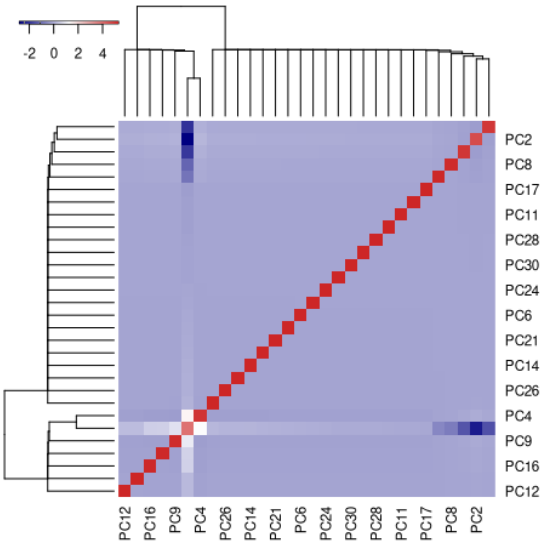

FIGURE 7. Corrloation heatmap for Principal Component Loadings

It can be seen that most of the principal components are not really correlated, however between PC1 and PC9, there is a lot of corrolation going on. By looking at the values in the matirx used to create this heatmap, we find that the corrolation between PC1 and PC2 is -0.5906, which is the highest and then for PC1 and PC3 it is roughly -0.7372. Interestingly, PC1 and PC4 are highly positively corrolated, coming in at a value of around 0.31.

Considering the above, before performing PCA-LDA I am making the choice to use 6 principal components for the following analysis. This is further supported by using *summary(tumPCAScaled)* which shows us that the first 6 principle components make up 90% of the total variance.

3. LDA AND L-O-O

We run through LDA in the standard way, as can be seen in the code given in the appendix.

	B	M
B	356	0
M	29	180

Finally, runnign *sum(diag(prob.table(class)))* which gives us a finally accuracy of 95% on classification with 6 principal components.

Appendices

R Source Code

```
tumourData = read.csv("/home/matthew/Documents/University/PDS/Datasets/tumour.csv")

#Create a correlation matrix, plot heatmaps
corMat = as.matrix(cor(tumourData[,3:32]))
heatmap3(corMat)
heatmap3(corMat,scale = "none")

#PCA Analysis

tumPCA = prcomp(tumourData[,3:32])
biplot(tumPCA)

boxplot(tumourData[,3:32]) #Suggests scaling

tumPCAScaled = prcomp(tumourData[,3:32], scale = T)
biplot(tumPCAScaled, cex = 0.6)

#Stitch together new dataset and leave out outliers
tumDataNew = rbind(tumourData[1:3,],tumourData[5:152,],tumourData[154:212,],tumourData[2

tumPCAFinal =prcomp(tumDataNew[,3:32], scale=T)
biplot(tumPCAFinal, cex= 0.6)

corMatPCA <- as.matrix(cor(tumPCAFinal$rotation))
heatmap3(corMatPCA)

#LDA/L-O-O

pcaScores = tumPCAFinal$x[,1:6]
tumLDA = lda(pcaScores, tumDataNew[,2], CV=T)
table(tumDataNew[,2], tumLDA$class)

class=table(tumDataNew[,2],tumLDA$class)
diag(prop.table(class,1))
sum(diag(prop.table(class)))
```