Assignment 3 STAT 394

Irshad Ul Ala

20200913

Contents

1 Principal Component Analysis (PCA) of the wine dataset

```
wine <- read.csv("D:/Uni/STAT 394/wine(1).csv")
```

1.1 Changing Class into a factor

```
str(wine)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Class              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol            : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic_acid         : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash                : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity_of_ash  : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium          : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Total_phenols       : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids          : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavonoid_phenols: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins     : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color_intensity     : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue                 : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ Diluted_wines       : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline             : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

A quick examination of the dataset shows that importing the data has by default defined the 3 classes as an integer, which we change with the factor function.

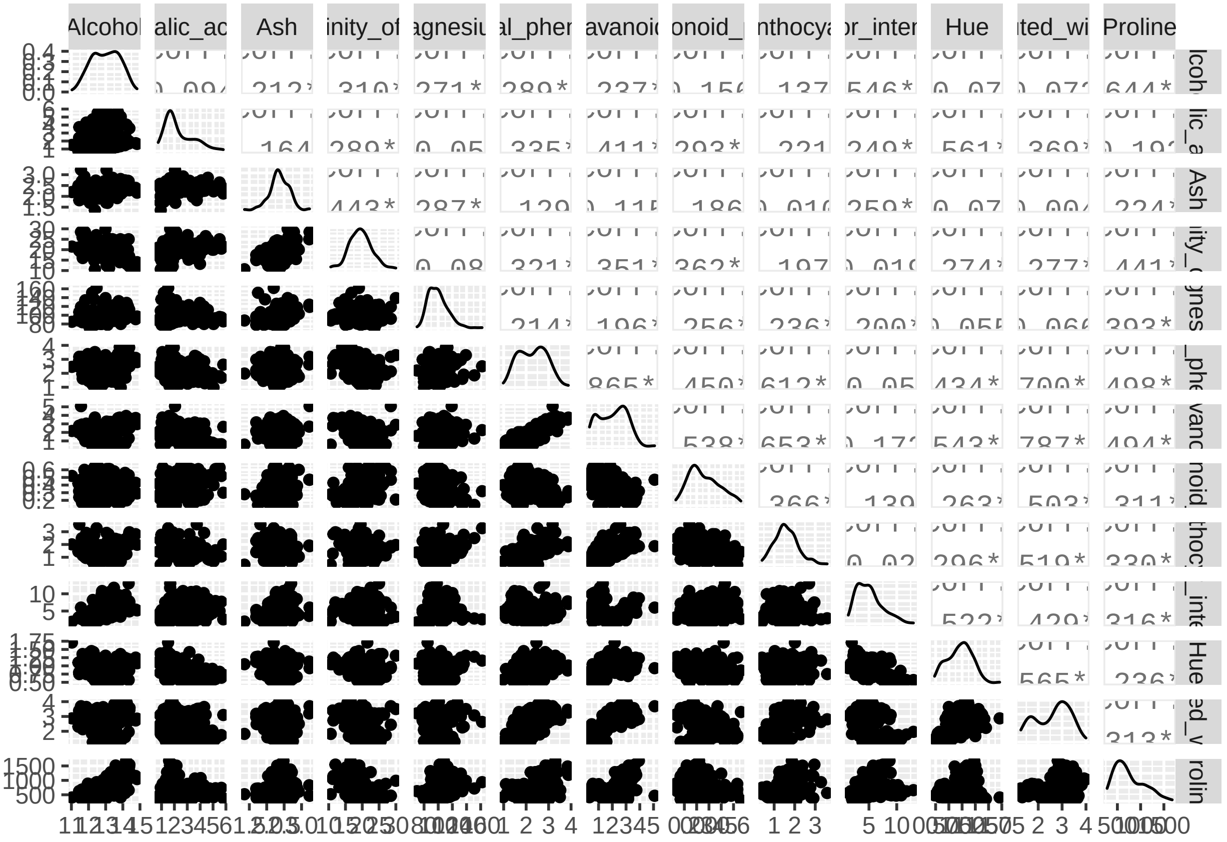
```
wine$Class <- factor(wine$Class,levels = c("1","2","3"),labels=c("Class 1","Class 2","Cl
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Class               : Factor w/ 3 levels "Class 1","Class 2",..: 1 1 1 1 1 1 1 1 1
##  $ Alcohol             : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic_acid          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash                 : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity_of_ash   : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium           : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Total_phenols       : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids          : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavonoid_phenols: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins     : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color_intensity     : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue                 : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ Diluted_wines       : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline             : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

1.2 Correlation between pairs of the numerical variables

We begin by examining if the wine dataset has numerical variables which demonstrate significant correlation. If there is significant collinearity between variables, we should consider a principal component analysis, and observe what linear combinations of the numerical variables would constitute orthogonal sets of axes that are linearly independent from each other, implying zero correlation between "sets' of numerical variables.

```
ggpairs(wine[,-1])
```



1.3 Identifying significant correlations >0.7

Using the scale function, the wine dataset is copied and standardized ('normalised') so that we can form a correlation matrix directly from the covariance matrix function.

Following that, we will filter out the correlation values that are above a magnitude of 0.7, to get an idea of the number of pairs of variables with high correlation.

```
winecopy<-scale(wine[,-1])
titles<-names(wine)[-1]

for(i in 1:13){
for(j in 1:13){
  cor = cov(winecopy)[i,j]
  if(abs(i-j)>0){
    if(abs(cor)>=0.7){
      print(cor)
      print(titles[i])
      print('correlation with')
      print(titles[j])
      print("---")
      }
    }
  }
}
```

```
## [1] 0.8645635
## [1] "Total_phenols"
## [1] "correlation with"
## [1] "Flavanoids"
## [1] "---"
## [1] 0.8645635
## [1] "Flavanoids"
## [1] "correlation with"
## [1] "Total_phenols"
## [1] "---"
## [1] 0.7871939
## [1] "Flavanoids"
## [1] "correlation with"
## [1] "Diluted_wines"
## [1] "---"
## [1] 0.7871939
## [1] "Diluted_wines"
## [1] "correlation with"
## [1] "Flavanoids"
## [1] "---"
```

Due to the nature of covariance

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] = E[(Y - \mu_Y)(X - \mu_X)] = Cov(Y, X)$$

,the matrix is a symmetric matrix, and as such we will be finding the same high correlation twice. In other words, despite there having been 4 located correlations with a correlation coefficient magnitude above 0.7, there are in reality, 4/2=2 pairs of variables with high correlation with each other: Flavonoids with Diluted wines, and Flavonoids with Total Phenols.

1.4 Applying Principal Component Analysis to dataframe

1.4.1 The Principal Axes

Removing the first 'Class' column, we apply principal component analysis. We begin by examining what the principal axes would look like. The coefficients per row of each Principal Component(PC) vector indicate the coefficient required to form the linearly independent eigenvector.

```
PCAwp<-prcomp(wine[,-1], center=TRUE, scale=TRUE)
options(digits=2)
PCAwp$rotation[,1:8]
```

```
##                          PC1     PC2    PC3    PC4    PC5    PC6    PC7    PC8
## Alcohol              -0.1443  0.4837 -0.207  0.018 -0.266  0.214 -0.056  0.396
## Malic_acid            0.2452  0.2249  0.089 -0.537  0.035  0.537  0.421  0.066
## Ash                   0.0021  0.3161  0.626  0.214 -0.143  0.154 -0.149 -0.170
## Alcalinity_of_ash     0.2393 -0.0106  0.612 -0.061  0.066 -0.101 -0.287  0.428
## Magnesium            -0.1420  0.2996  0.131  0.352  0.727  0.038  0.323 -0.156
## Total_phenols        -0.3947  0.0650  0.146 -0.198 -0.149 -0.084 -0.028 -0.406
## Flavanoids           -0.4229 -0.0034  0.151 -0.152 -0.109 -0.019 -0.061 -0.187
## Nonflavonoid_phenols  0.2985  0.0288  0.170  0.203 -0.501 -0.259  0.595 -0.233
## Proanthocyanins      -0.3134  0.0393  0.149 -0.399  0.137 -0.534  0.372  0.368
## Color_intensity       0.0886  0.5300 -0.137 -0.066 -0.076 -0.419 -0.228 -0.034
## Hue                  -0.2967 -0.2792  0.085  0.428 -0.174  0.106  0.232  0.437
## Diluted_wines        -0.3762 -0.1645  0.166 -0.184 -0.101  0.266 -0.045 -0.078
## Proline              -0.2868  0.3649 -0.127  0.232 -0.158  0.120  0.077  0.120
```

Due to the number of variables, it is difficult to recognise special eigenvectors (like an eigenvector indicating the 'mean' and so forth). However, some basic observations can be made, such as the fact that the variable Ash and Color Intensity, are not significant contributors to forming the first eigenvector.

In contrast, Total Phenols and Alcalinity of Ash become unimportant variables to constructing the second eigenvector.

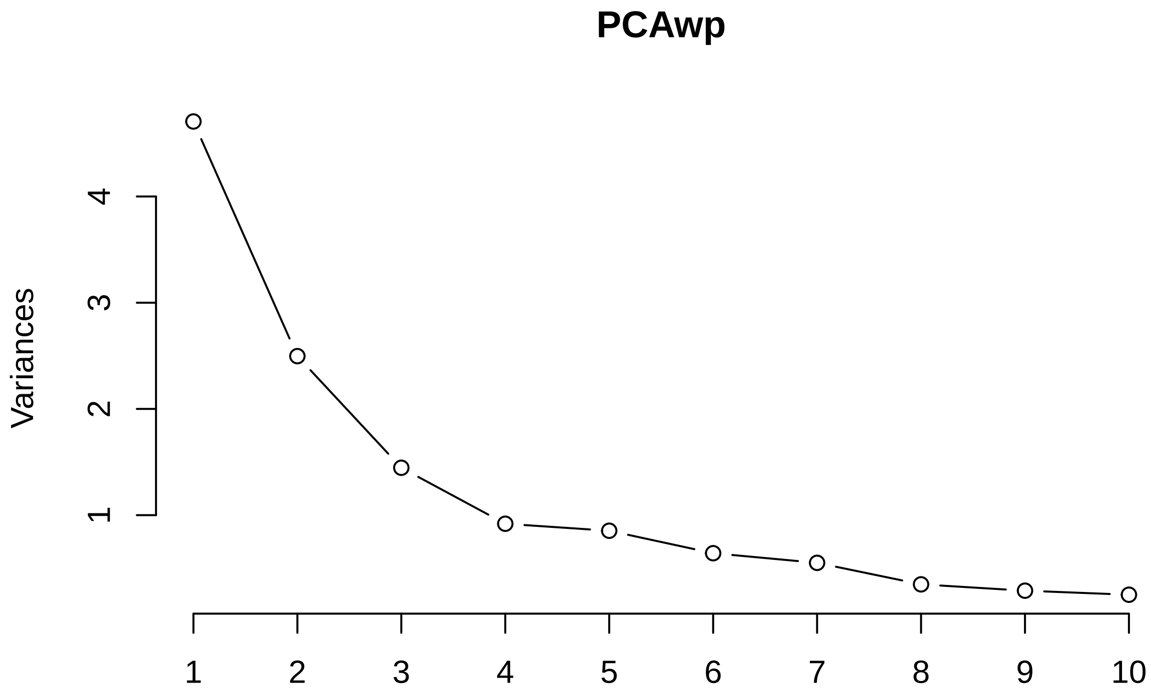
However, it is also important to quantify the significance of each of the principal component axes in explaining the variation in the model.

1.4.2 Amount of Variation explained by each principal component axis/eigenvector

```
summary(PCAwp)
```

```
## Importance of components:
##                          PC1   PC2   PC3    PC4    PC5    PC6    PC7    PC8
## Standard deviation     2.169 1.580 1.203 0.9586 0.9237 0.8010 0.7423 0.5903
## Proportion of Variance 0.362 0.192 0.111 0.0707 0.0656 0.0494 0.0424 0.0268
## Cumulative Proportion  0.362 0.554 0.665 0.7360 0.8016 0.8510 0.8934 0.9202
##                           PC9   PC10   PC11  PC12    PC13
## Standard deviation     0.5375 0.5009 0.4752 0.411 0.32152
## Proportion of Variance 0.0222 0.0193 0.0174 0.013 0.00795
## Cumulative Proportion  0.9424 0.9617 0.9791 0.992 1.00000
```

```
plot(PCAwp,type="l")
```

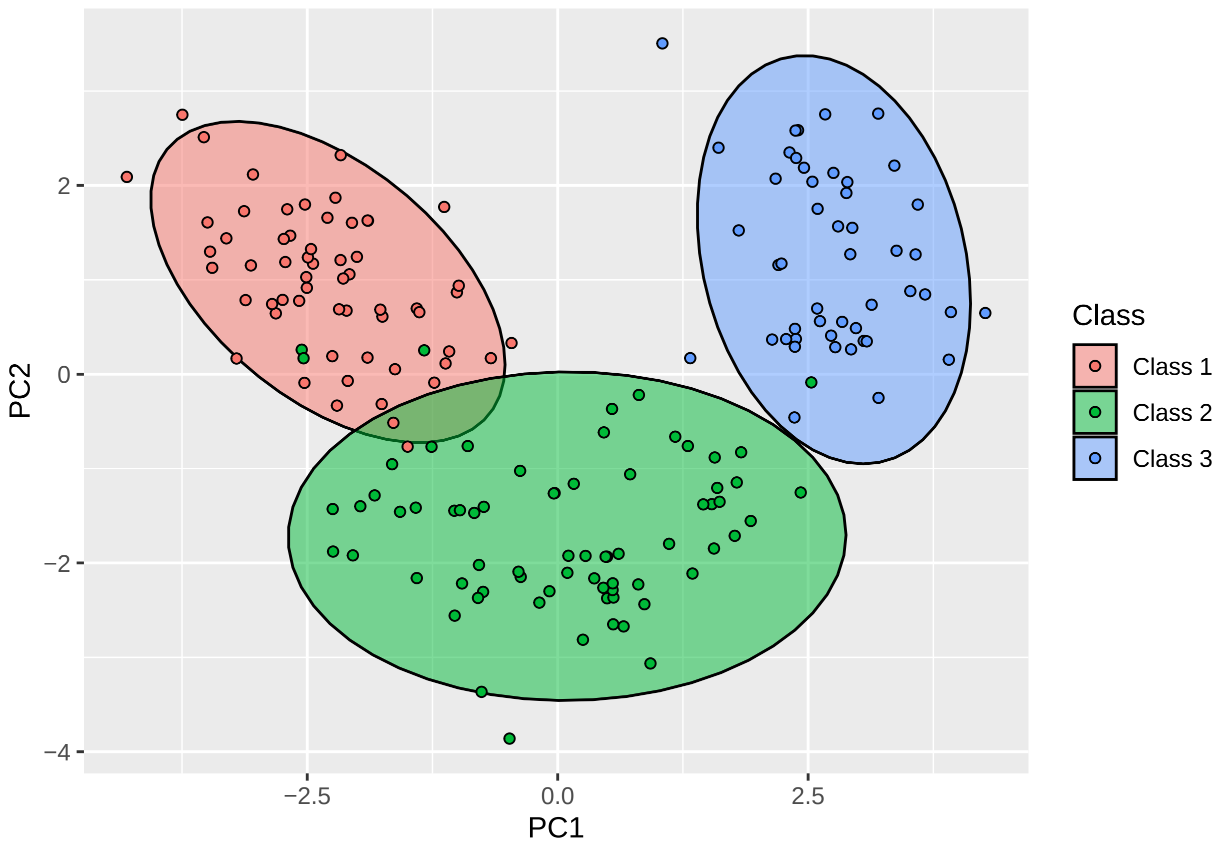


Unlike previous cases, it is evident that the first eigenvector does not explain an exponentially larger amount of variation compared to the other axes. By design, it inevitably explains the most, but at more than 50% effectiveness, the 2nd principal axis, PC2, is able to explain the variation of the data points (19.2% for PC2 and 36.2% for PC1). This trend appears to continue all the way to the final principal axis, PC13.

1.5 Visualisations

1.5.1 Effectiveness of PC1 and PC2 in distinguishing classes visually

```
newdata<-cbind(wine, PCAwp$x[,1:2])
ggplot(newdata, aes(PC1, PC2, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")
```



The resultant seems demonstrate clearly how the first 2 principal axes separate the data points from one another, possibly distinguishing classes from one another.

1.5.2 Correlations between variances and PCs

To understand how relevant each of the variables were to framing PC1 and 2 for instance, we can form a correlation matrix describing that phenomenon.

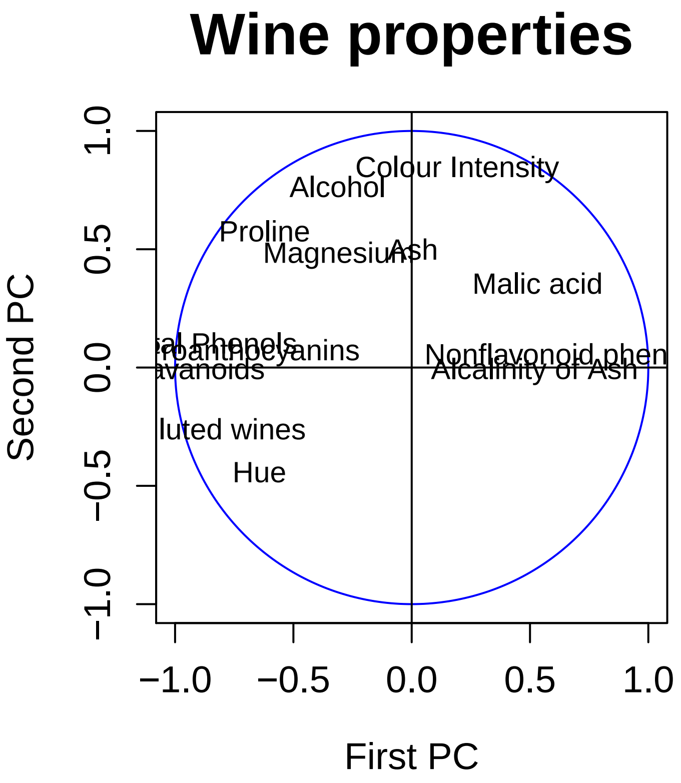
```
cor(wine[,-1],PCAwp$x[,1:2])
```

```
##                          PC1     PC2
## Alcohol              -0.3131  0.7643
## Malic_acid            0.5319  0.3554
## Ash                   0.0044  0.4994
## Alcalinity_of_ash     0.5192 -0.0167
## Magnesium            -0.3080  0.4735
## Total_phenols        -0.8561  0.1028
## Flavanoids           -0.9175 -0.0053
## Nonflavonoid_phenols  0.6476  0.0455
## Proanthocyanins      -0.6799  0.0621
## Color_intensity       0.1922  0.8375
## Hue                      -0.6437 -0.4412
## Diluted_wines            -0.8160 -0.2599
## Proline                  -0.6221  0.5766
```

It is worth nothing that this set of results are **NOT** the same as the the eigenvectors of the principal axes earlier. These results demonstrate how much correlated the variables were with the principal axes themselves. We will find later that the later principal axes are not nearly as involved in cementing that distinction between the classes.

1.5.3 Plotting the correlations

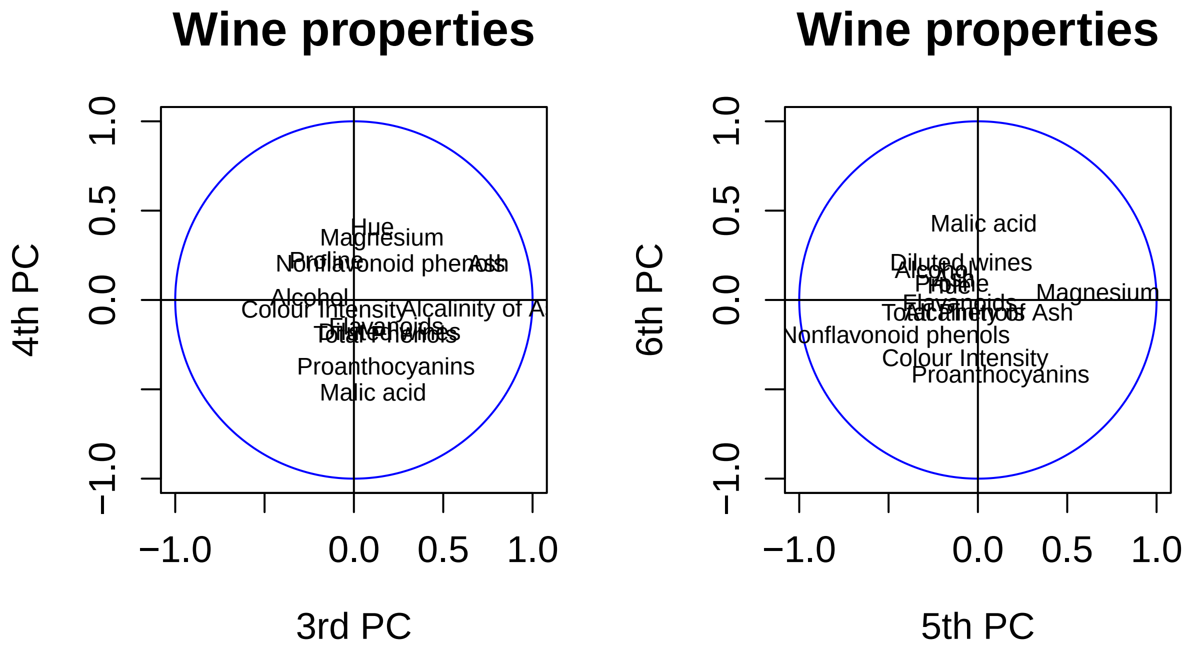
```
r1<-cor(wine[,-1],PCAwp$x[,1:2])

par(pty="s")
ucircle=cbind(cos((0:360)/180*pi ) , sin((0:360)/180*pi))
plot(ucircle, type="l",lty="solid", col="blue", lws = 2, xlab="First PC", ylab="Second P
abline(h=0.0, v=0.0)

text(x=r1, label=c("Alcohol", "Malic acid", "Ash", "Alcalinity of Ash", "Magnesium", "To
```



The variables close to the periphery of the circle indicate that their variances are well explained by the first 2 principal axes. In this case, Total phenols, Flavanoids, Diluted wins and Proanthocyanins are the best explained variables by the first 2 principal axes. In contrast, Alcalinity of Ash appears to be the least well explained variable. Despite that, it is still a decently explained variable.

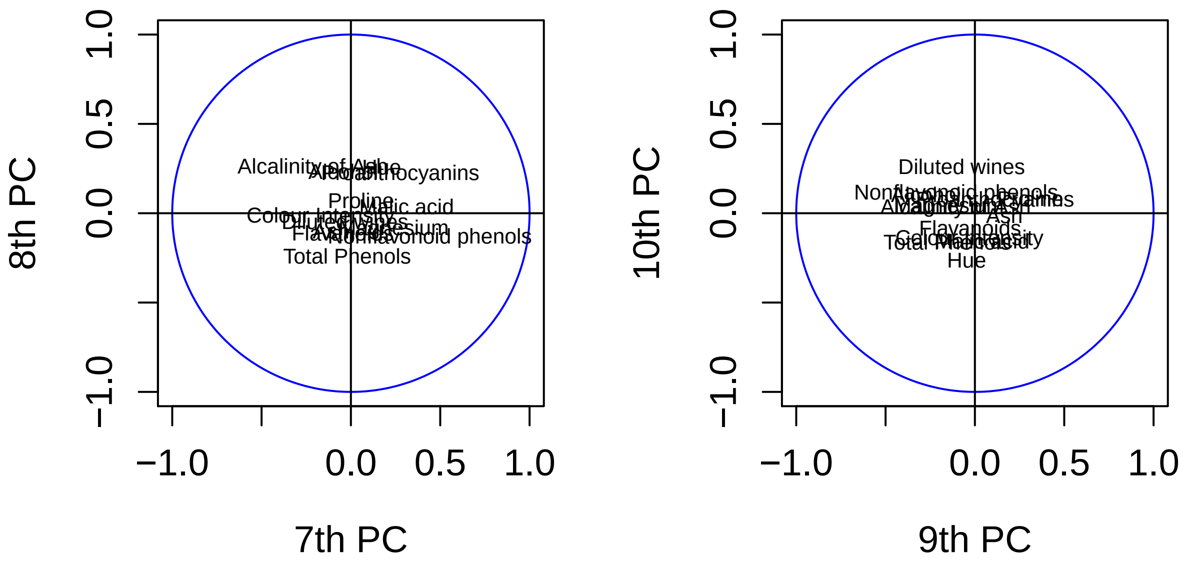
For later principal axes however, the variables of interest appear to change.

```
par(pty="s")
par(mfrow = c(1,2))
r1<-cor(wine[,-1],PCAwp$x[,3:4])
ucircle=cbind(cos((0:360)/180*pi ) , sin((0:360)/180*pi))
plot(ucircle, type="l",lty="solid", col="blue", lws = 2, xlab="3rd PC", ylab="4th PC", m
abline(h=0.0, v=0.0)
text(x=r1, label=c("Alcohol", "Malic acid", "Ash", "Alcalinity of Ash", "Magnesium", "To
r1<-cor(wine[,-1],PCAwp$x[,5:6])
ucircle=cbind(cos((0:360)/180*pi ) , sin((0:360)/180*pi))
plot(ucircle, type="l",lty="solid", col="blue", lws = 2, xlab="5th PC", ylab="6th PC", m
abline(h=0.0, v=0.0)
text(x=r1, label=c("Alcohol", "Malic acid", "Ash", "Alcalinity of Ash", "Magnesium", "To
```



For instance, in the case of the 3rd and 4th PC, Alcalinity of Ash's variance is best explained, while that of Flavanoids and Colour intensity were the least well explained by this second pair of principal axes. For the 3rd pair of principal axes (PC 5 and 6), Malic acid's variance, which was previously mediocrely represented by the previous 2 pairs of principal axes, appears to be the relatively best explained.

```
par(pty="s")
par(mfrow = c(1,2))
r1<-cor(wine[,-1],PCAwp$x[,7:8])
ucircle=cbind(cos((0:360)/180*pi ) , sin((0:360)/180*pi))
plot(ucircle, type="l",lty="solid", col="blue", lws = 2, xlab="7th PC", ylab="8th PC",
abline(h=0.0, v=0.0)
text(x=r1, label=c("Alcohol", "Malic acid", "Ash", "Alcalinity of Ash", "Magnesium", "To
r1<-cor(wine[,-1],PCAwp$x[,9:10])
ucircle=cbind(cos((0:360)/180*pi ) , sin((0:360)/180*pi))
plot(ucircle, type="l",lty="solid", col="blue", lws = 2, xlab="9th PC", ylab="10th PC",
abline(h=0.0, v=0.0)
text(x=r1, label=c("Alcohol", "Malic acid", "Ash", "Alcalinity of Ash", "Magnesium", "To
```

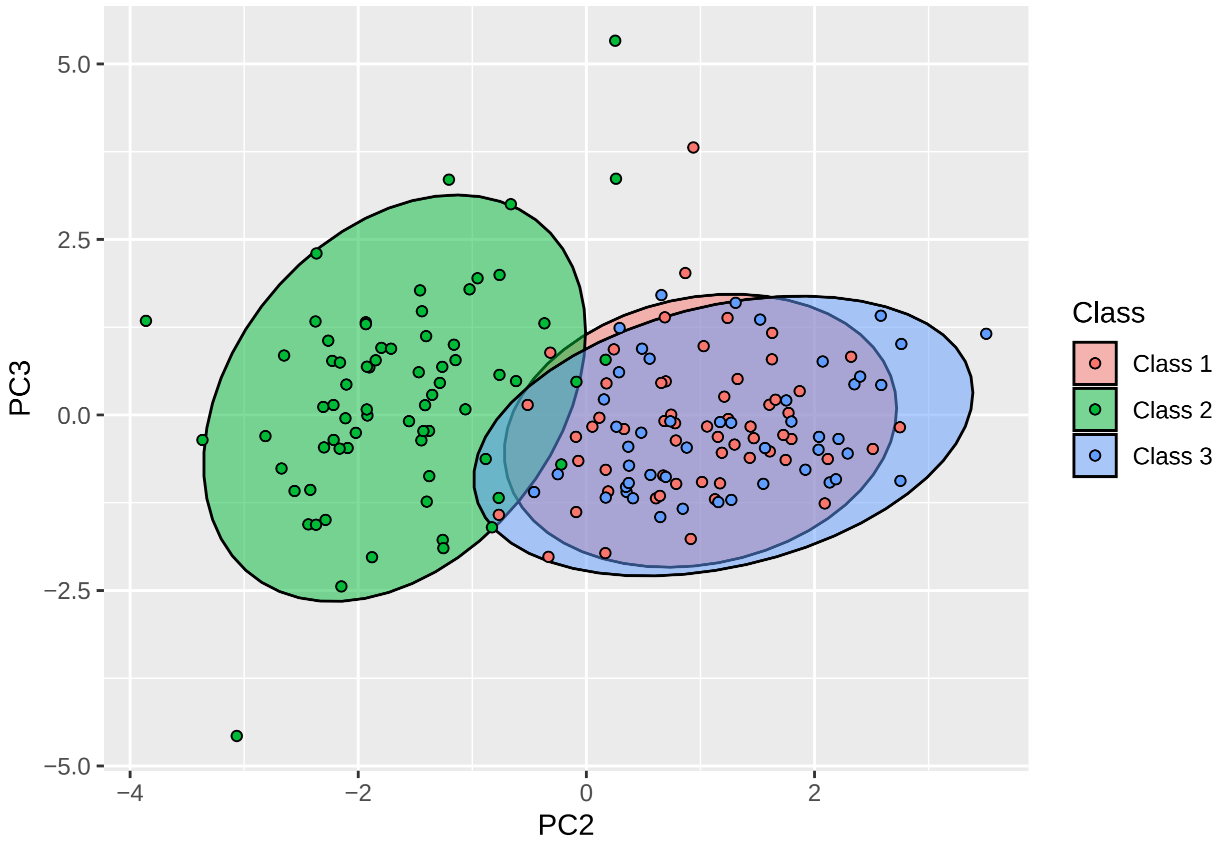


As expected, as we go down the list of principal axes, the amount of variance explained by them in general decreases and very few variables appear to be favoured by the later principal axes. Diluted wines and Hue appear to be relatively better explained by the 9th and 10th principal axes, regardless. There are definitely limitations to this biplot visualisation method which are especially evident here. The sheer number of explanatory variables makes it somewhat difficult to account for in these plots.

For the sake of curiosity, the extent to which the later principal axes appear to differentiate the points between classes can be plotted.

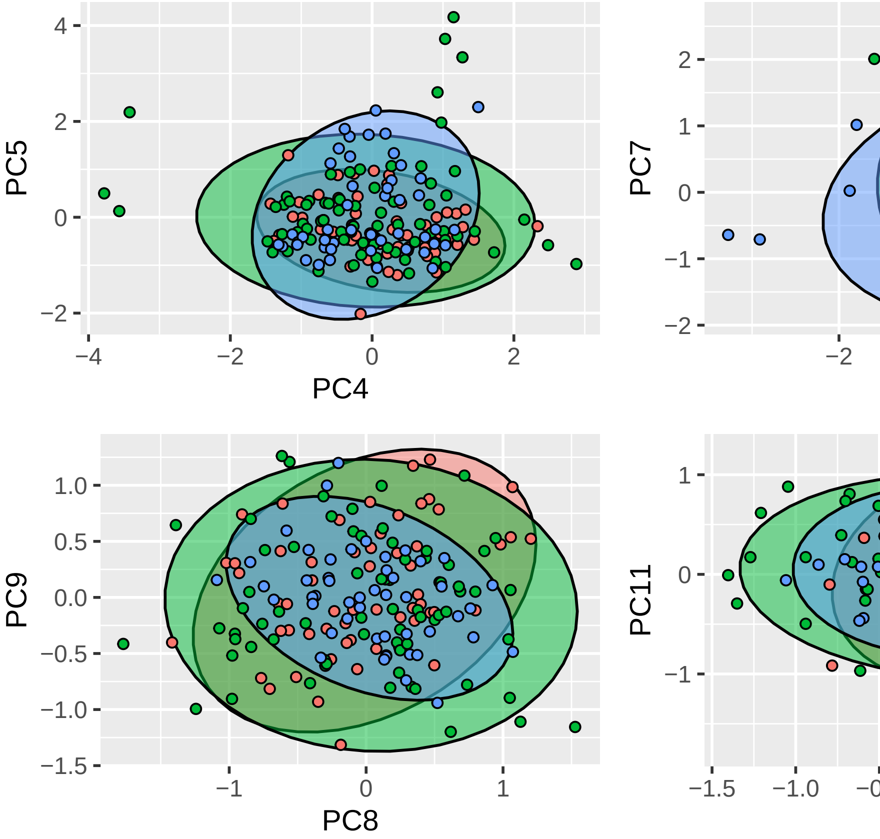
1.5.4 *Effectiveness of later applied principal axes in distinguishing classes visually

```r
newdata1<-cbind(wine, PCAwp$x[,1:13])

ggplot(newdata1, aes(PC2, PC3, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")
```



As evidenced by the fact that the classes are not vertically distinguishable from one another, PC3 appears to not be involved in differentiating the data points from one another based on class but on other characteristics.

```
plot1<-ggplot(newdata1, aes(PC4, PC5, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")+theme(legend.position="none")
plot2<-ggplot(newdata1, aes(PC6, PC7, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")+theme(legend.position="none")
plot3<-ggplot(newdata1, aes(PC8, PC9, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")+theme(legend.position="none")
plot4<-ggplot(newdata1, aes(PC10, PC11, col=Class, fill=Class))+
stat_ellipse(geom="polygon", col = "black", alpha=0.5)+
geom_point(shape=21, col="black")+theme(legend.position="none")

grid.arrange(plot1, plot2,plot3, plot4, ncol=2)
```



1.5.4.1 Later principal axis plots

As expected, the plots indicate that class differentiation is not of concern for the later axes(PC3 onwards).