Identify the Typology of Wine through the Chemical Attributions Using Multivariate Analysis

Yi Chen

Final Paper for Multivariate Analysis I (HUDM 6122)

May 2020

Introduction

This project will show the example of using R to carry out some simple multivariate analyses, with a focus on explortary data analysis (unsupervised learning) and classification analysis (supervised learning).

For the explortary data analysis, we will focus on the principal component analysis (PCA). For the cluster analysis, we will compare the results of linear discriminate analysis (LDA) and quadratic discriminatie analysis (QDA).

Whole project will guide the reader from discriptive data analysis to more advanced statistical methodologies. For every technique covered in this project, the corresponding research question and method theory (including assumption and limitation). R code and outcome will be provided as we explore these techniques.

Data Set

The data sets in this project come from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml). In particular, the wine recognition data (http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data) is updated at Sept 21, 1998 by C.Blake. The original source of the data set comes from Forina, M. et al (1991).

The observations in the data set are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. In total, there are 178 observations.

```
## load the data

wine <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",sep=",")
colnames(wine) <- c("type", "Alcohol","Malic acid","Ash","Alcalinity","Magnesium","Phenols","Flavanoids","Nonflavanoid ph
enols","Proanthocyanins","Color intensity","Hue","OD280/OD315","Proline")
head(wine,5)

##   type Alcohol Malic acid  Ash Alcalinity Magnesium Phenols Flavanoids
## 1    1   14.23       1.71 2.43       15.6       127    2.80       3.06
## 2    1   13.20       1.78 2.14       11.2       100    2.65       2.76
## 3    1   13.16       2.36 2.67       18.6       101    2.80       3.24
## 4    1   14.37       1.95 2.50       16.8       113    3.85       3.49
## 5    1   13.24       2.59 2.87       21.0       118    2.80       2.69
##   Nonflavanoid phenols Proanthocyanins Color intensity  Hue OD280/OD315 Proline
## 1                 0.28            2.29            5.64 1.04       3.92    1065
## 2                 0.26            1.28            4.38 1.05       3.40    1050
## 3                 0.30            2.81            5.68 1.03       3.17    1185
## 4                 0.24            2.18            7.80 0.86       3.45    1480
## 5                 0.39            1.82            4.32 1.04       2.93     735

dim(wine)

## [1] 178  14
```

Descriptive Data Analysis

```
wine$type <- factor(wine$type)
summary(wine)
```

```
##  type      Alcohol         Malic acid        Ash             Alcalinity
##  1:59   Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60
##  2:71   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20
##  3:48   Median :13.05   Median :1.865   Median :2.360   Median :19.50
##         Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49
##         3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50
##         Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00
##    Magnesium       Phenols        Flavanoids     Nonflavanoid phenols
##  Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
##  1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
##  Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
##  Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
##  3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
##  Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600
##  Proanthocyanins  Color intensity       Hue          OD280/OD315       Proline
##  Min.   :0.410    Min.   : 1.280   Min.   :0.4800   Min.   :1.270   Min.   : 278.0
##  1st Qu.:1.250    1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5
##  Median :1.555    Median : 4.690   Median :0.9650   Median :2.780   Median : 673.5
##  Mean   :1.591    Mean   : 5.058   Mean   :0.9574   Mean   :2.612   Mean   : 746.9
##  3rd Qu.:1.950    3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0
##  Max.   :3.580    Max.   :13.000   Max.   :1.7100   Max.   :4.000   Max.   :1680.0
```

As we can see, the first column of the data set is *type*, which indicates which wine type the sample belongs to. The 13 distinct attributes are: *Alcohol*, *Malic acid*, *Ash*, *Alcalinity*, *Magnesium*, *Phenols*, *Flavanoids*, *Nonflavanoid phenols*, *Proanthocyanins*, *Color intensity*, *Hue*, *OD280/OD315*, and *Proline*. There is no missing value in the data set. All of 13 attributes are continuous variable with different scale. For example, the mean of proline is 746.9 and mean of hue is 0.965. The standard deviation of each attribute are also very different.

```
round(apply(wine[2:14],MARGIN = 2, sd),2)
```

```
##          Alcohol       Malic acid              Ash
##             0.81             1.12             0.27
##       Alcalinity        Magnesium          Phenols
##             3.34            14.28             0.63
##       Flavanoids Nonflavanoid phenols     Proanthocyanins
##             1.00             0.12             0.57
##  Color intensity              Hue        OD280/OD315
##             2.32             0.23             0.71
##          Proline
##           314.91
```

This indicates that there is a need for normalize the data before the analysis. Meanwhile, scale the data also reshape the distribution of the data so that it is closer to the normal distribution, which is the assumption for many multivariate analysis models.

Mean difference across wine type after normalization

According to the analysis, I first normalized the data set, which make all 13 attributes have the same standard deviation as 1. Then, we can compare the different of each attribute. For example, three group have clear difference in the average level of *alcohol*. Type 2 wine is the only one has the average negative value of alcohol. Similar pattern shows in all attributes. These results indicate that the attributes do have a strength in classifying the wine type. In the output below, I highlight the distinct patter for each attribute. These results partially indicate that these attributes are able to classify the wine type.

```
printz_scoreByGroup <- function(variables,groupvariable){
    means <- aggregate(as.matrix(variables) ~ groupvariable, FUN = mean)
    means$groupvariable <- as.numeric(means$groupvariable)
    print(round(means,2))
}

printz_scoreByGroup(scale(wine[2:14]),wine$type)
```

groupvariable	Alcohol	Malic acid	Ash	Alcalinity	Magnesium	Phenols
1	0.92	-0.29	0.32	**-0.74**	**0.46**	**0.87**
2	**-0.89**	-0.36	**-0.44**	0.22	-0.36	-0.06
3	0.19	**0.89**	0.26	0.58	-0.03	-0.98

groupvariable	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	D280/OD315	Proline
1	0.95	**-0.58**	0.54	0.20	0.46	0.77	**1.17**
2	0.05	0.01	0.07	**-0.85**	0.43	0.24	-0.72
3	**-1.25**	0.69	**-0.76**	1.01	**-1.20**	**-1.31**	-0.37

Between-groups Variance and Within-groups Variance for a Variable

Further, we can calculate the within group variance, between group variance, and Separation of each variable. Similar to the idea in ANOVA, this information helps us to identify which variable is more useful in classify the wine type and which varible is more consistant across different wine type in general.

The within group variance can be calculated using the formula:

$$\frac{\sum(y_{i1}-\overline{y_1})^2+\ldots+\sum(y_{g1}-\overline{y_g})^2}{n-g}$$

, where y_{ij} is the jth observation in the ith group, $\overline{y_i}$ is the average of the ith group, g is the number of groups, n is the number of total samples.

```
WithinGroupsVariance <- function(variable,groupvariable){
    n_total <- length(variable)
    n_group <- length(unique(groupvariable))
    total_variance <- 0
    for (g in 1:n_group){
      group_data <- variable[groupvariable==g]
      group_mean <- mean(group_data)
      for (i in group  data){
        total_variance <- total_variance + (i - group_mean)^2
      }
```

```
    }
    return(total_variance/(n_total-n_group))
}
WithinGroupsVariance(wine$Alcohol,wine$type)

## [1] 0.2620525
```

As the example, the within group variande for *alcohol* is about 0.26. Simiarly, we can calculate the between group variance using the formula below,

$$\frac{n_1(\overline{y_1}-\overline{y})^2+\ldots+n_g(\overline{y_g}-\overline{y})^2}{g-1}$$

, where n_i represent the number of samples in group i, $\overline{y_i}$ is the group mean, and $\bar{y}$ is the overall average of all observation.

```
BetweenGroupsVariance <- function(variable,groupvariable){
    n_group <- length(unique(groupvariable))
    grand_mean <- mean(variable)
    total_variance <- 0
    for (g in 1:n_group){
        group_data <- variable[groupvariable==g]
        group_mean <- mean(group_data)
        ng <- length(group_data)
        total_variance <- total_variance + ng*(group_mean - grand_mean)^2
    }
    return(total_variance/(n_group-1))
}
BetweenGroupsVariance(wine$Alcohol,wine$type)

## [1] 35.39742
```

As the example, the within group variande for *alcohol* is about 35.39. We can calculate the "separation" achieved by a variable as its between-groups variance devided by its within-groups variance. The separation for *alcohol* is consequently 135.07 (i.e., 35.39742/0.2620525).

The separation indicate that much more variance in the data set can be explianed by the group difference than the within group uncertainty. A high value of separation means the attribute is useful in classfication since the variance difference across group is significant. Similarly, we can calculate all variables in the data set.

```
for (i in 2:14){
  between <- BetweenGroupsVariance(wine[,i],wine$type)
  within <- WithinGroupsVariance(wine[,i],wine$type)
  variablename <- colnames(wine)[i]
  sep <- between / within
  print(paste(variablename,",Wv=",within,",Bv=",between,",Sep=",sep))
}

## [1] " Alcohol,       Wv= 0.26,      Bv= 35.39,      Sep= 135.07"
## [1] " Malic acid,    Wv= 0.88,      Bv= 32.78,      Sep= 36.94"
## [1] " Ash,           Wv= 0.06,      Bv= 0.87,       Sep= 13.31"
## [1] " Alcalinity,    Wv= 8.00,      Bv= 286.41,     Sep= 35.77"
## [1] " Magnesium,  Wv= 180.65,   Bv= 2245.50,   Sep= 12.42"
## [1] " Phenols,                 Wv= 0.19,       Bv= 17.92 ,     Sep= 93.73"
## [1] " Flavanoids,              Wv= 0.27,       Bv= 64.26,      Sep= 233.92"
## [1] " Nonflavanoid phenols,  Wv= 0.01,       Bv= 0.32,       Sep= 27.57"
## [1] " Proanthocyanins ,      Wv= 0.24,       Bv= 7.45,       Sep= 30.27"
```

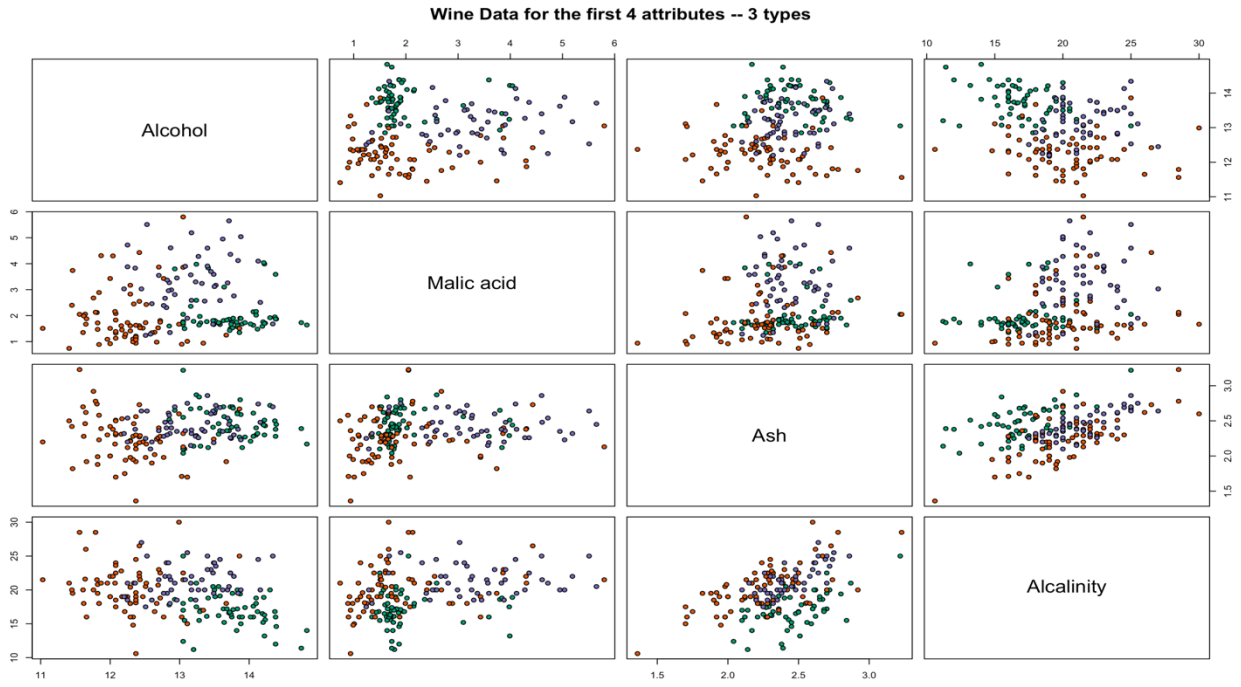
```
## [1] " Color intensity,          Wv= 2.28,       Bv= 275.70 ,      Sep= 120.66"
## [1] " Hue,                      Wv= 0.02,       Bv= 2.48,         Sep= 101.31"
## [1] " OD280/OD315,              Wv= 0.16,       Bv= 30.54,        Sep= 189.97"
## [1] " Proline,                  Wv= 29707.68,  Bv= 6176832.32, Sep= 207.92"
```

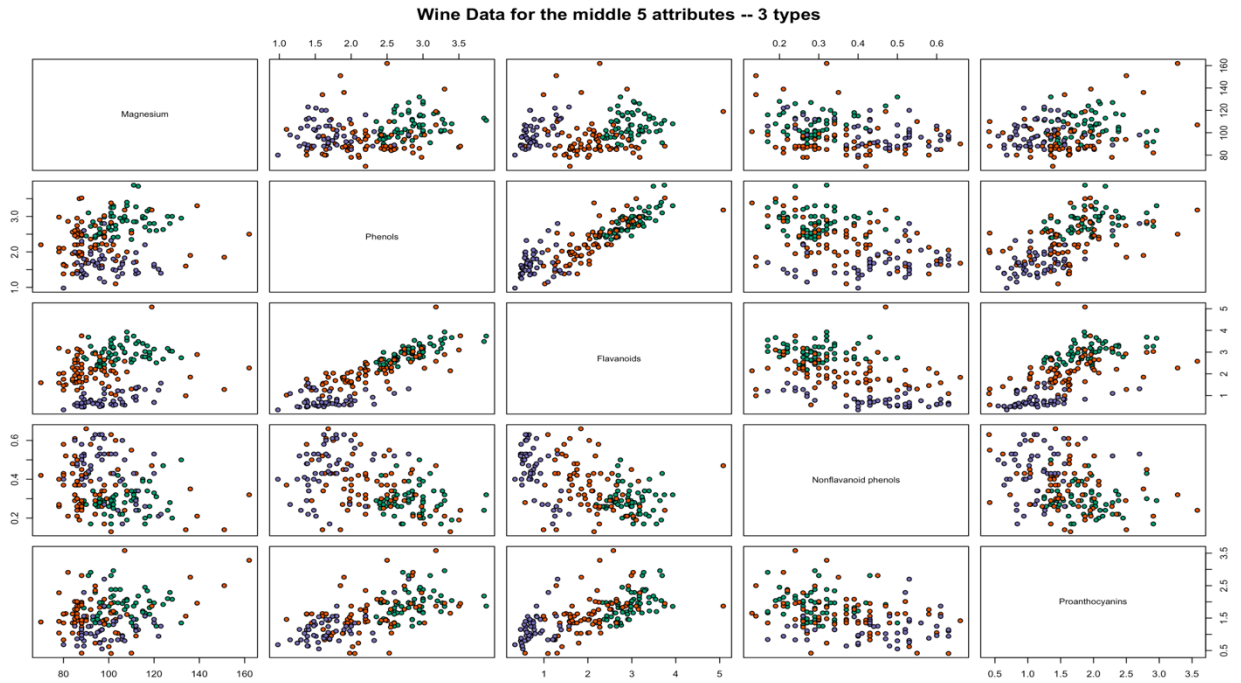
As we can see from the analysis, the variable *Flavanoids* and *Proline* have the high separation, while *Magnesium* and *Ash* have the low separation.

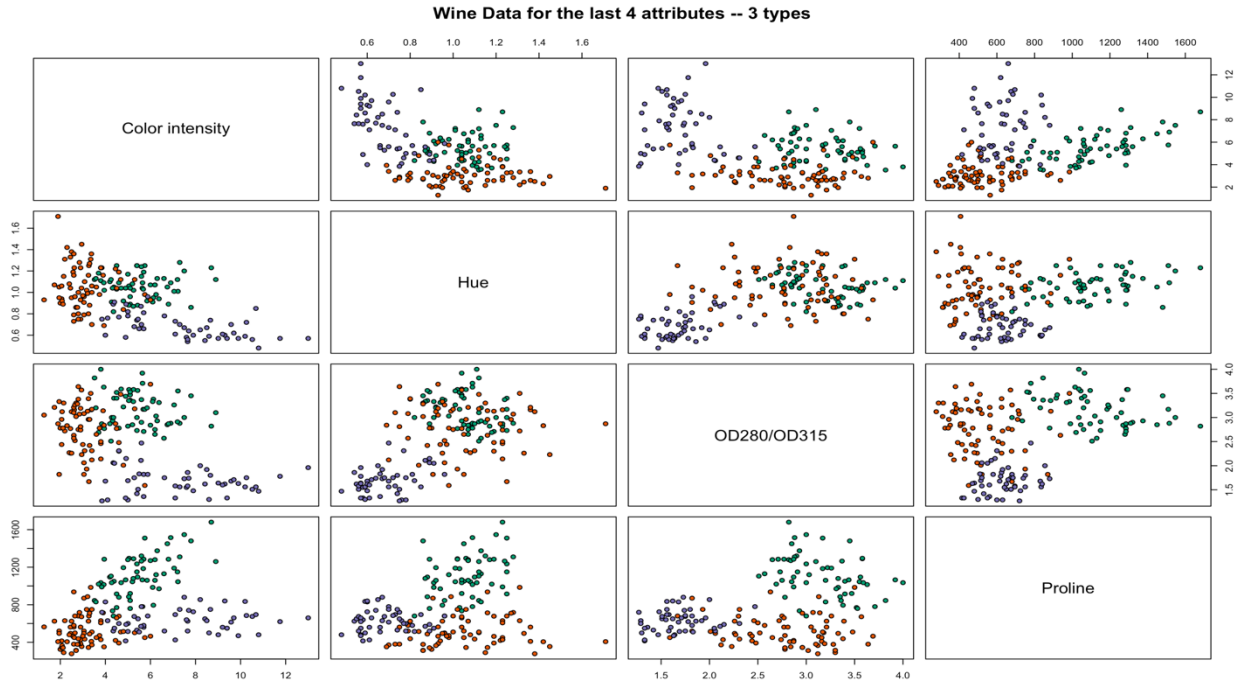
Matrix Scatter Plot

In multivariate analysis, a common way to understand the structure of the data set through visualization is matrix scatterplot.

```
library(car)
library(ggplot2)
pairs(wine[2:5],
    main = "Wine Data for the first 4 attributes -- 3 types",
    pch = 21,
    bg = c("#1b9e77", "#d95f02", "#7570b3")
  [unclass(wine$type)])

pairs(wine[6:10],
    main = "Wine Data for the middle 5 attributes -- 3 types",
    pch = 21,
    bg = c("#1b9e77", "#d95f02", "#7570b3")
    [unclass(wine$type)])

pairs(wine[11:14],
    main = "Wine Data for the last 4 attributes -- 3 types",
    pch = 21,
    bg = c("#1b9e77", "#d95f02", "#7570b3")
    [unclass(wine$type)])
```



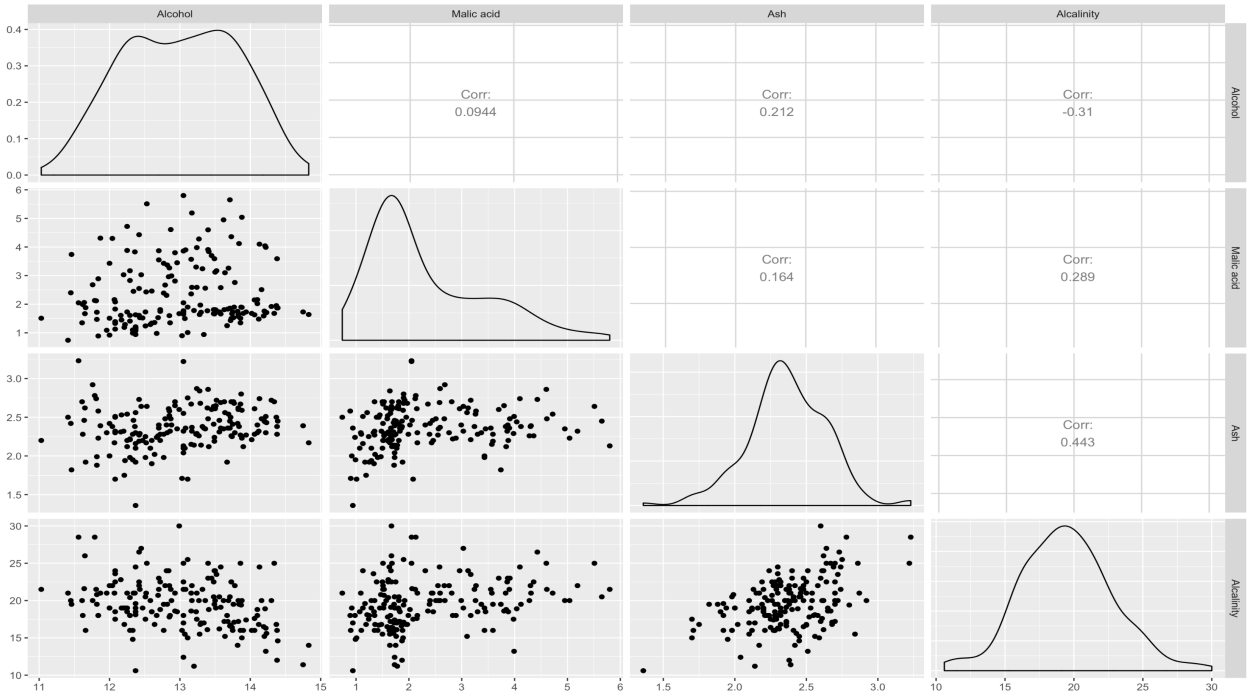


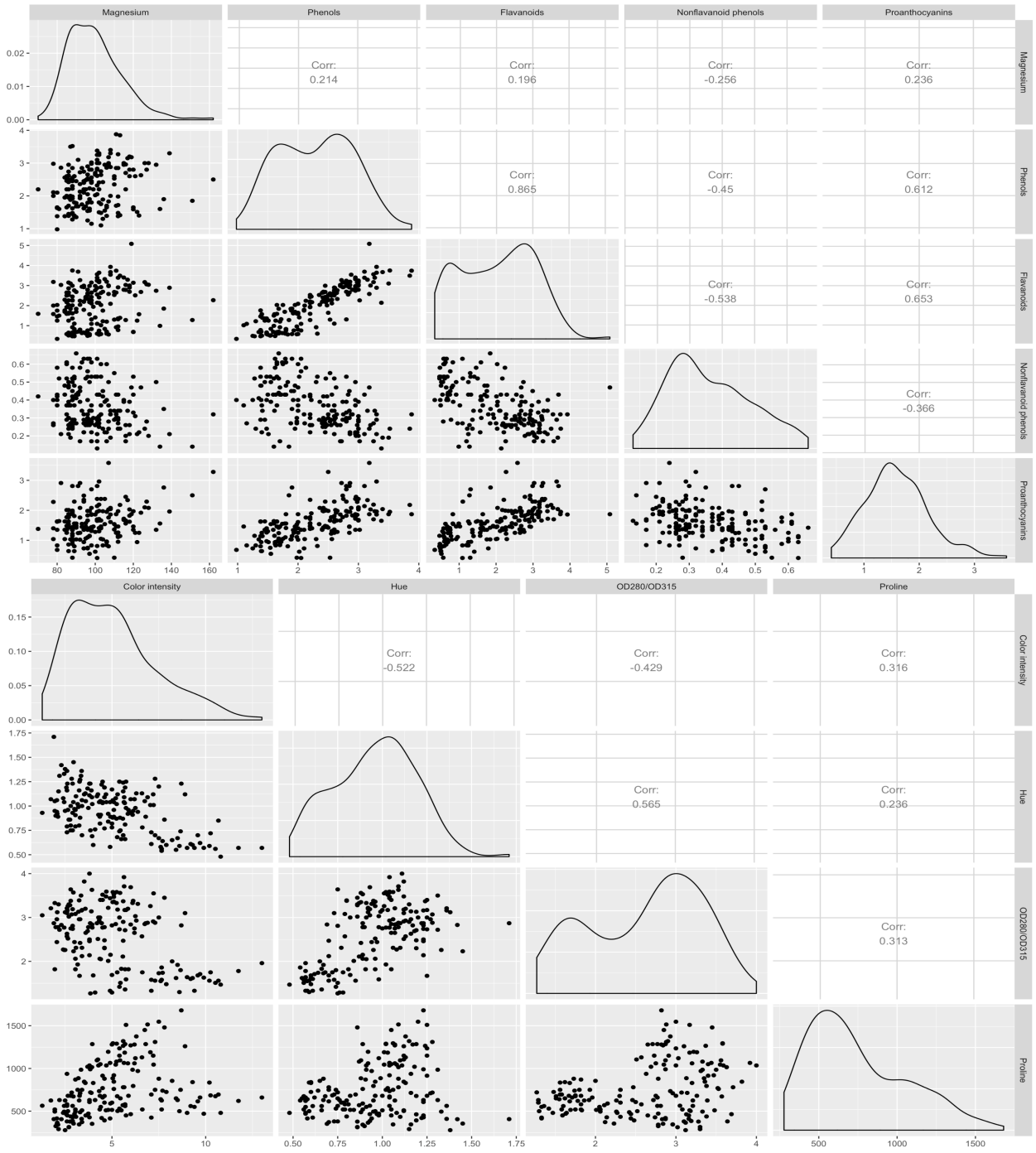


Using the scatter plot give us a more detail information about how each pair of two attributes are correlated with each other. The color in the scatter plots indicates the type of wine. In general, there are already patterns in classification for most of pairwise scatterplots. For example, as the *phenols* and *flavonoids* both increases, the type of wine change correspondingly. These results indicate that these attributes will be useful to classify the wine type.

Final Paper

```
library(GGally)
ggpairs(wine[2:5])

ggpairs(wine[6:10])

ggpairs(wine[11:14])
```





In the matrix scatterplot above, the diagonal cells show distribution of each of the variables. Each of the off-diagonal cells is a scatterplot of two of the attributes and the corresponding correlation coefficients. In terms of normality, most of the atrribute are approximately follow the normal distribution. However, attributs like alcohol, phenols, flavanoids, OD280/OD315, and profile do have clear difference from the normal distribution. In terms of correlation, most of the correlation are not significant. However, the correlation between flavanoids and phenols is the only one above 0.8.

Principal Component Analysis

Summary of PCA

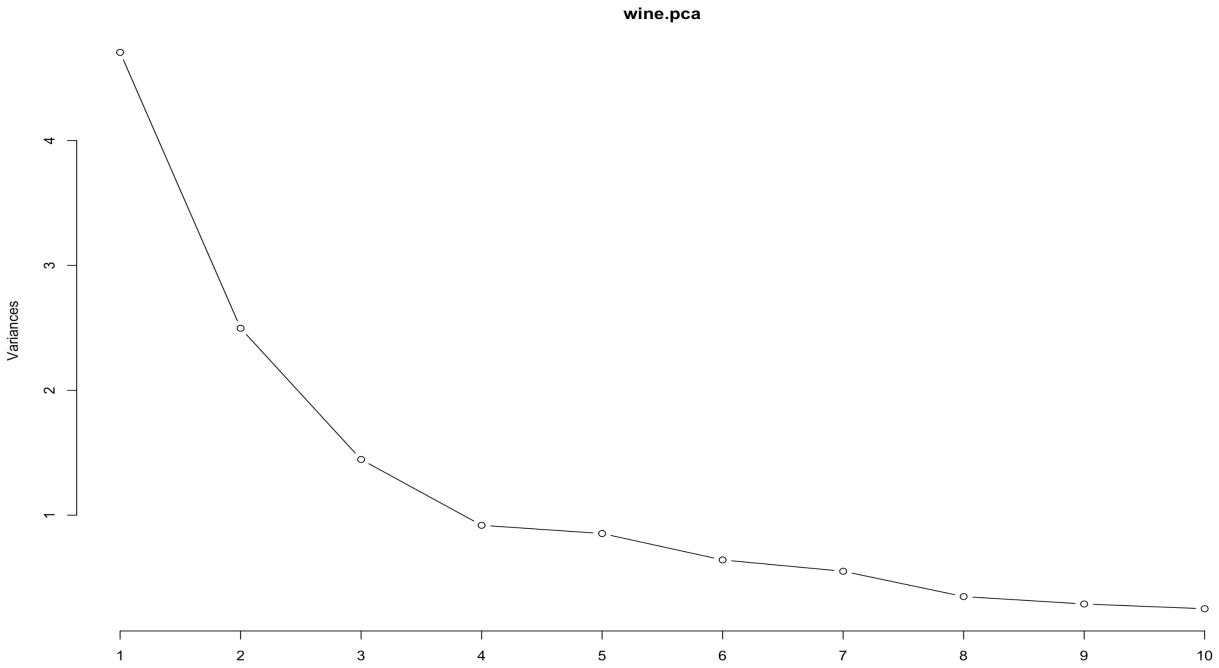
Principal component analysis (PCA) simplifies the complexity in high-dimensional data while retaining trends and patterns. In other word, PCA is a technique for feature extraction — so it combines input variables in a specific way, then we can drop the "least important" variables while still retaining the most valuable parts of all of the variable. It reduce the high dimension of data space into a lower dimension principal component space through orthogonal linear transformation. The new coordinate system created by the principaal component has greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

The biggest assunmption of PCA is the linear assumption. PCA can find the orthogonal projections of the dataset that contains the highest variance if the data set is linear correlated. Otherwise, it cannot help to reduce the dimension. In other word, the underlying structure of the data must be linear, patterns that are highly correlated may be unresolved because all PCs are uncorrelated, and the goal is to maximize variance and not necessarily to find clusters. Second, PCA rely on the orthogonal tranformations. Howeever, it may not be the best data transformation in terms of extracting the feature. Third, PCA is scale variant. This means that the scale of the data set will change the result of the PCA. Finally, PCA assume no missing value in the data set.

PCA Data Analysis

To carry out a principal component analysis (PCA) on a multivariate data set, the first step is often to standardise the variables. This is necessary if the input variables have very different variances, which is true in this case as the concentrations of the 13 chemicals have very different variances (see above).

```
scale_imput <-scale(wine[,2:14])
wine.pca <- prcomp(scale_imput)
summary(wine.pca)

## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.169  1.5802 1.2025 0.95863 0.92370 0.80103 0.74231
## Proportion of Variance 0.362  0.1921 0.1112 0.07069 0.06563 0.04936 0.04239
## Cumulative Proportion  0.362  0.5541 0.6653 0.73599 0.80162 0.85098 0.89337
##                           PC8     PC9    PC10    PC11    PC12    PC13
## Standard deviation     0.59034 0.53748 0.5009 0.47517 0.41082 0.32152
## Proportion of Variance 0.02681 0.02222 0.0193 0.01737 0.01298 0.00795
## Cumulative Proportion  0.92018 0.94240 0.9617 0.97907 0.99205 1.00000

screeplot(wine.pca, type="lines")
```



The most obvious change in slope in the scree plot occurs at component four, which is the "elbow" of the scree plot. Therefore, it cound be argued based on the basis of the scree plot that the first three components should be retained. As we can see the first four principal component keeps about 73.59% variance. There are some other rules to we can pick to determine the number of principal components to keep.

```
cumsum((wine.pca$sdev)^2)/sum((wine.pca$sdev)^2)
```

```
##  [1] 0.3619885 0.5540634 0.6652997 0.7359900 0.8016229 0.8509812 0.8933680
##  [8] 0.9201754 0.9423970 0.9616972 0.9790655 0.9920479 1.0000000
```

For example, if we need to ensure that at least 80% of the total variance should be maintained. Thus, first five principal components should be picked. Another rule is to pick the principal component which has the variance bigger than 1. In this way, we only need to keep the first three principal component.

```
(wine.pca$sdev)^2
```

```
##  [1] 4.7058503 2.4969737 1.4460720 0.9189739 0.8532282 0.6416570 0.5510283
##  [8] 0.3484974 0.2888799 0.2509025 0.2257886 0.1687702 0.1033779
```

T Take the first principal as an example, the loading is shown as below.

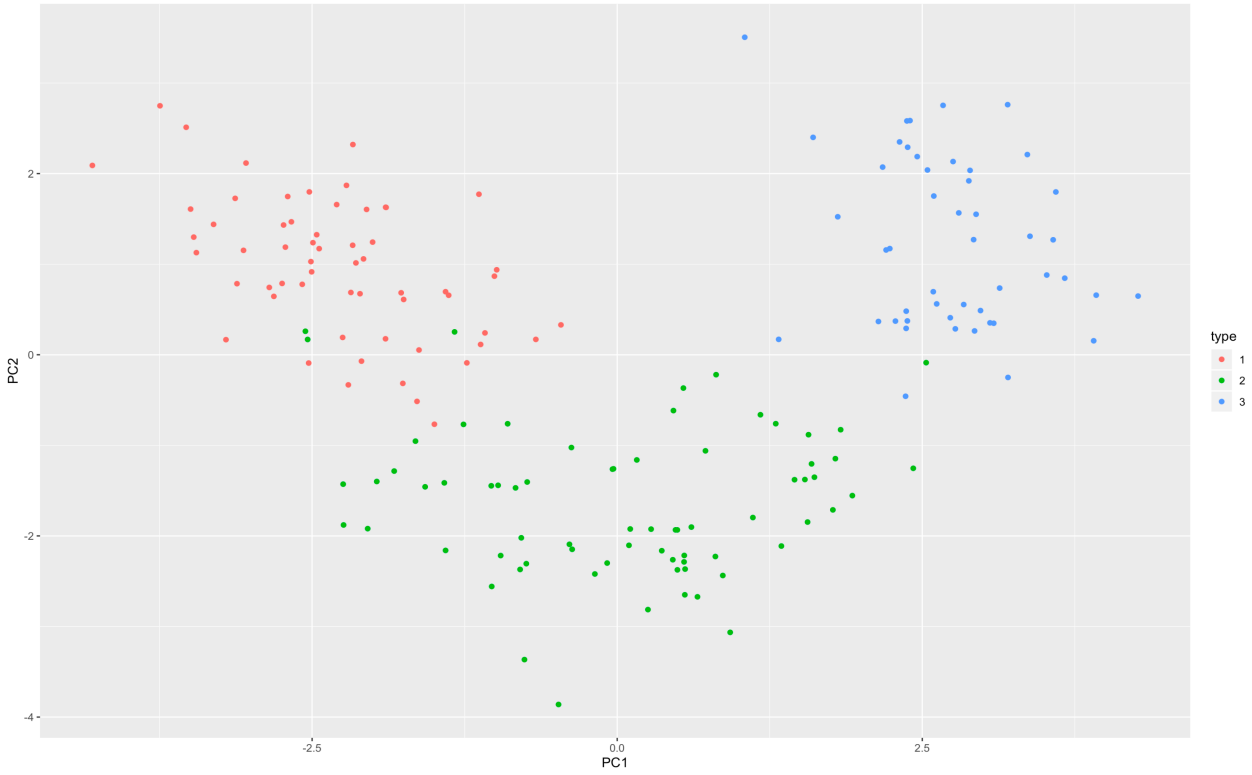
```
round(wine.pca$rotation[,1],2)
```

```
##            Alcohol          Malic acid                 Ash
##              -0.14               0.25               0.00
##         Alcalinity           Magnesium            Phenols
##               0.24              -0.14              -0.39
##         Flavanoids Nonflavanoid phenols      Proanthocyanins
##              -0.42               0.30              -0.31
##    Color intensity                 Hue     OD280/OD315          Proline
##               0.09              -0.30              -0.38            -0.29
```

This indicates that the first principal component relies more on *Flavanoids*, *Phenols*, and *OD280/OD315*. The variable *Ash* almost contribute zero to the first principal component. The loading of the principal component is helpful in interpretating the possible meaning of the principal component. In this study, I am not familiar with chemical knowledge of these attributes. So, I will not take the risk to summary the meaning of the principal component.

Finllay, scatterplot of the first two principal components, and label the data points with the cultivar that the wine samples come from will be take as an example.

```
plot_data <- data.frame('PC1'=wine.pca$x[,1],'PC2'=wine.pca$x[,2],type=wine$type)
ggplot(plot_data,aes(x=PC1,y=PC2)) +
  geom_point(aes(col=type))
```



The scatterplot shows the first principal component on the x-axis, and the second principal component on the y-axis. We can see from the scatterplot that wine samples of cultivar 1 have much lower values of the first principal component than wine samples of cultivar 3. Therefore, the first principal component separates wine samples of cultivars 1 from those of cultivar 3. We can also see that wine samples of cultivar 2 have much higher values of the second principal component than wine samples of cultivars 1 and 3. Therefore, the second principal component separates samples of cultivar 2 from samples of cultivars 1 and 3. Therefore, the first two principal components are reasonably useful for distinguishing wine samples of the three different cultivars. If we add more principal components into the analysis, we can expect that the classification will have higher accuracy.

Linear Discriminate Analysis

Summary of LDA and QDA

Linear discriminate analysis (LDA) is a method to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

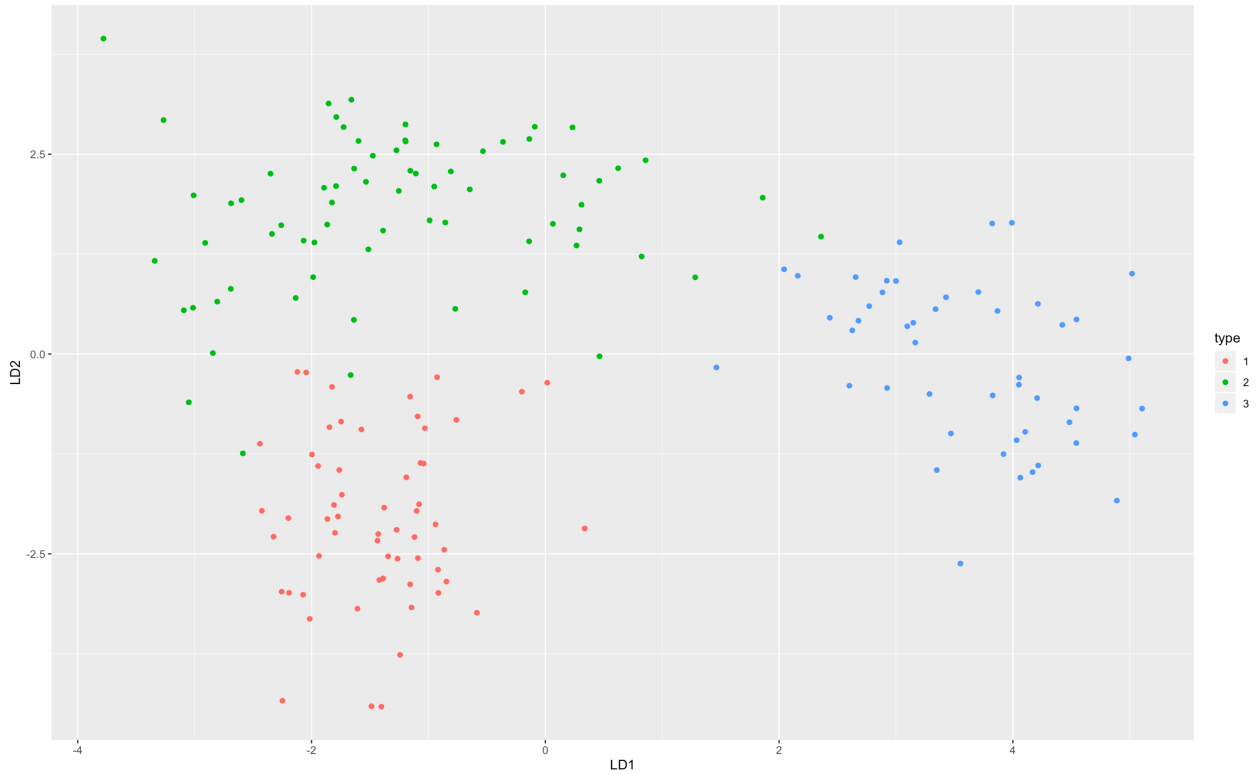
LDA assumes that the approaches the problem by assuming that the conditional probability density functions from both classes are both normally distributed with differnt mean vector and the same variance-covariance matrix (can also be different). If the varianes are assumed to be the same, it is called homoscedasticity assumption. Besides, LDA also take the multicollinearity and indepdence assumption. Predictive power can decrease with an increased correlation between predictor variables. Meanwhile, simples are assumed to be randomly sampled, and a participant's score on one variable is assumed to be independent of scores on that variable for all other participants. Thus, if the data does not follow the multivariate normal distribution, the variables have high multicollinearlity or the samples are not independent, the result of LDA will be biased. Besides, the LDA also rely on the complete data set and binary class.

If we do not make the assuption of equal group variance, the analysis will be qudratic discriminate analysis (QDA). Consequently, the decision boundary of QDA is qudratic instead of linear.

LDA analysis and QDA analysis

If we want to separate the wines by cultivar, the wines come from three different cultivars, so the number of groups is 3, and the number of variables is four principal component I calculated in the last section. The maximum number of useful discriminant functions that can separate the wines by cultivar is the minimum of 3-1 and 4, and so in this case it is the minimum of 2 and 13, which is 2. The variance is assumed to be the same.

```
library(MASS)
wine.lda <- lda(x = wine.pca$x[,1:4], grouping = wine$type)
lda.plot.data <- as.data.frame(predict(wine.lda)$x)
lda.plot.data$type <- wine$type
ggplot(lda.plot.data, aes(LD1, LD2)) +
  geom_point(aes(color = type))
```

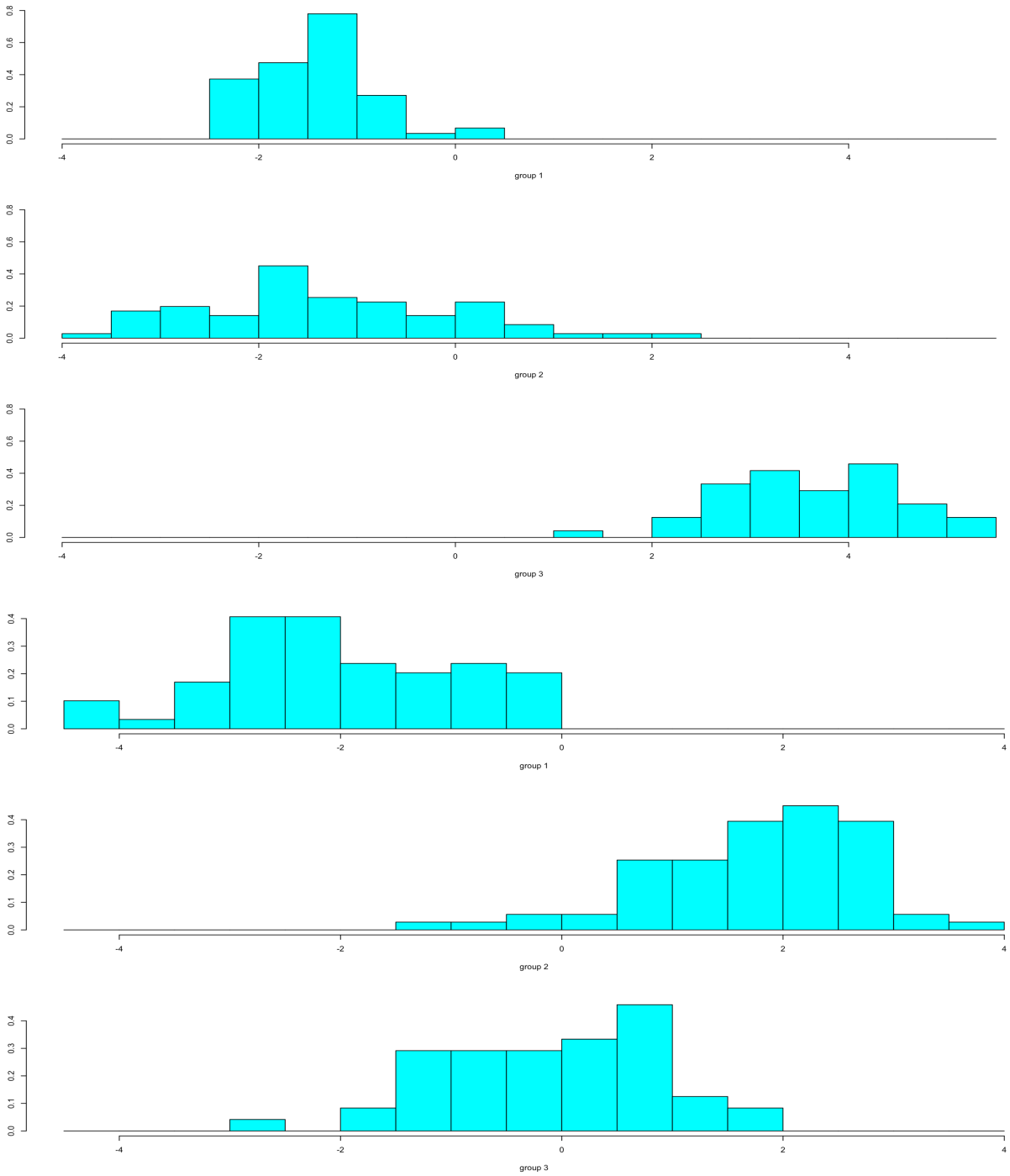


As we can see that the two discriminant functions separate the three-wine type well. The first discriminant function rely more on the first principal component, while the second discriminant function rely more on the second principal component.

```
wine.lda$scaling

##           LD1        LD2
## PC1  0.9606948  0.4427462
## PC2  0.7622075 -1.0176477
## PC3 -0.2018024  0.2830970
## PC4 -0.1167712 -0.1659482

table(Predicted=predict(wine.lda)$class, Actual=wine$type)

##                 Actual
## Predicted  1    2    3
##         1     59   3    0
##         2     0    66   0
##         3     0    2    48
```

As we can see from the table above, most of the data are predicted correctly. Only 2 points belongs to the range 3 are predicted to be in range 2. The overall aggrement rate is $1 - 5/178 = 97.19$. To futher analysis the separation achieved by each discriminant function. Stacked histogram can be used.

```
wine.pred <- predict(wine.lda, wine.pca$x[,1:4])
ldahist(data = wine.pred$x[,1], g=wine$type)

ldahist(data = wine.pred$x[,2], g=wine$type)
```



As we can see, the first discriminant function has the power to distinguish the type 1 and 2 with type 3. However, the difference between 1 and 2 are limiated. The differences between the 3 wine types are more clear seperated in the second discriminant function.

Similarly, we can do the QDA with the same data set and methods.

```
wine.qda <- qda(x = wine.pca$x[,1:4], grouping = wine$type)
table(Predicted=predict(wine.qda)$class, Actual=wine$type)
```

```
##                 Actual
## Predicted  1  2  3
##         1 59  1  0
##         2  0 69  0
##         3  0  1 48
```

As we can see QDA do imporve the overall model prediction. The overall aggrement rate is $1 - 2/178 = 98.87$. However, the LDA result is already good. I do not recommend to take the result of QDA to avoid the issue of model overfit.

Reference

Forina, M. et al. 1991. PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, *Via Brigata Salerno*, 16147 Genoa, Italy.