葡萄酒分类研究

无88 刘子源 2018010895
2020年5月21日

一、研究背景

葡萄酒的化学成分十分复杂，不同种类的葡萄酒的各种化学成分比例及含量是有差异的。通常检测葡萄酒的方法有感官评定和理化指标检测。感官评定有主观性强、干扰因素多、评测周期长、标准难以统一等缺点。常规的理化指标检测需要检测人员掌握大量的化学知识，应用复杂的化学分析技术，成本费用和劳力费用很高。是否存在着一种方法，只要知道了酒中所含的一些化学成分，就能在极短时间内判断出其种类？本研究基于多元统计分析方法，根据提供的葡萄酒的化学指标，来实现对其的自动分类。

二、数据分析

2.1数据介绍

该数据集包含了产于意大利同一地区但属于三个不同品种的葡萄酒的数据。数据有如下特征：样本178个，其中品种一样本60个，品种二样本69个，品种三样本49个，特征13个，都是通过化学分析得到的连续型数值，没有未知量。化学分析测定的成分属性分别为：

1）酒精（Alcohol）

2）苹果酸（Malic acid）

3）灰分（Ash）

4）灰分的碱度（Alkalinity of ash）

5）镁含量（Magnesium）

6）总酚（Total phenols）

7）黄酮类（Flavanoids）

8）非黄酮类苯酚（Nonflavanoid phenols）

9）原花青素（Proanthocyanins）

10）颜色强度（Color intensity）

11）色调（Hue）

12）稀释后的OD280/OD315比值（OD280/OD315 of diluted wines）

13）脯氨酸（Proline）

2.2探索性分析及预处理

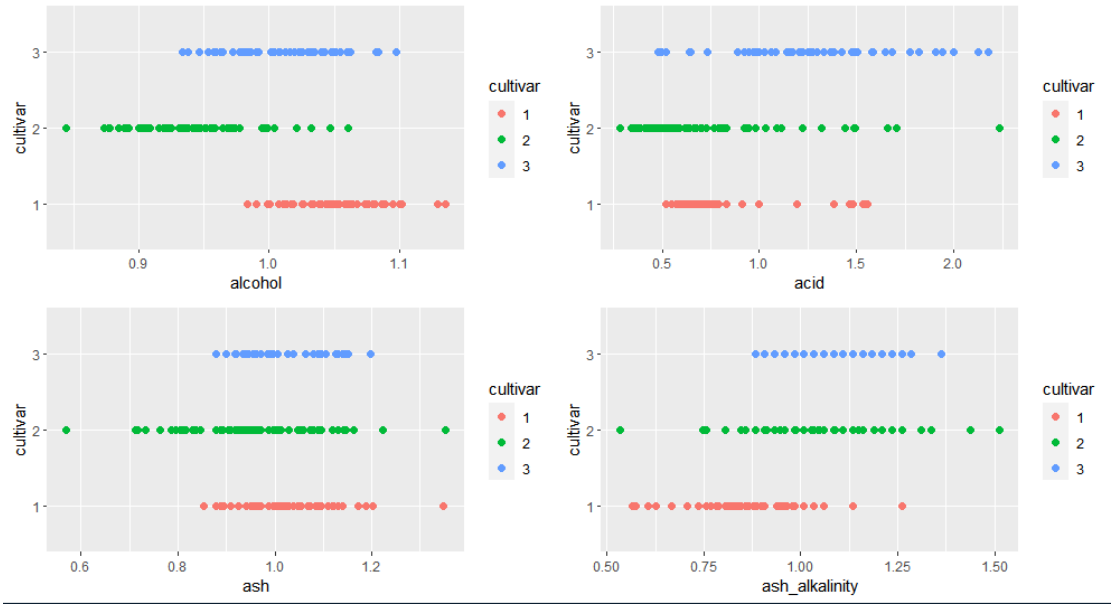
```
 cultivar    alcohol          acid            ash        ash_alkalinity         Mg
 1:59     Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00
 2:71     1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00
 3:48     Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00
          Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74
          3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00
          Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00
    phenols        flavanoids     nonflavanoid    proanthocyanins color_intensity
 Min.   :0.980   Min.   :0.340   Min.   :0.1300   Min.   :0.410   Min.   : 1.280
 1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700   1st Qu.:1.250   1st Qu.: 3.220
 Median :2.355   Median :2.135   Median :0.3400   Median :1.555   Median : 4.690
 Mean   :2.295   Mean   :2.029   Mean   :0.3619   Mean   :1.591   Mean   : 5.058
 3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375   3rd Qu.:1.950   3rd Qu.: 6.200
 Max.   :3.880   Max.   :5.080   Max.   :0.6600   Max.   :3.580   Max.   :13.000
      Hue           OD280_OD315       proline
 Min.   :0.4800   Min.   :1.270   Min.   : 278.0
 1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5
 Median :0.9650   Median :2.780   Median : 673.5
 Mean   :0.9574   Mean   :2.612   Mean   : 746.9
 3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0
 Max.   :1.7100   Max.   :4.000   Max.   :1680.0
```

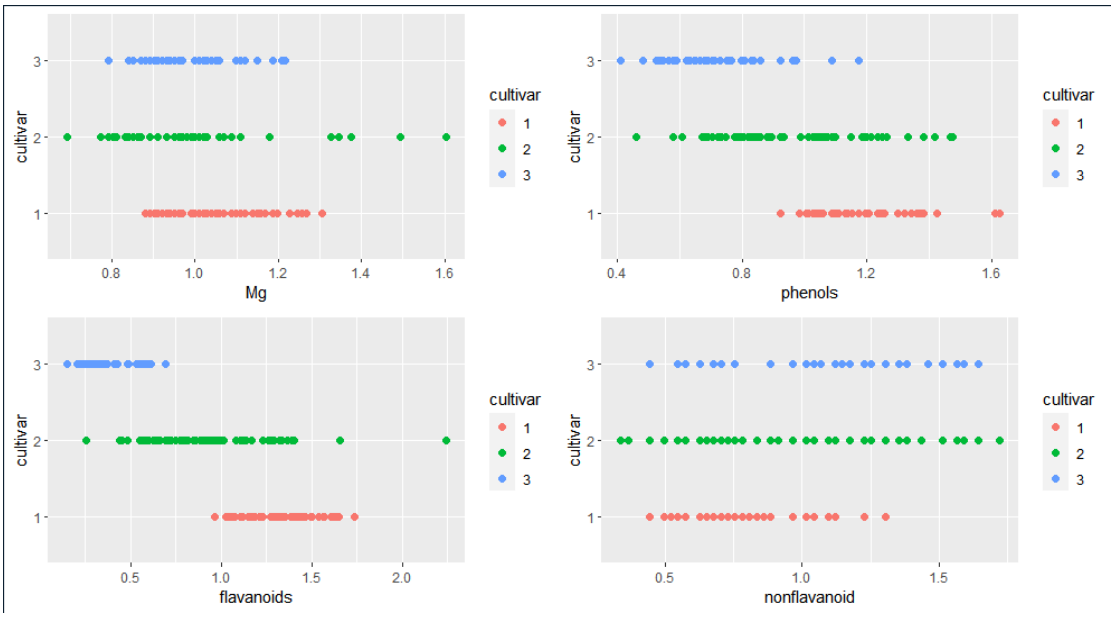
观察数据整体信息，发现变量Mg和proline的单位与其他变量明显不同，尤其是proline的数量级是其他变量的100到1000倍，若直接拿原始数据分析会使得数据的变异性几乎全部集中在proline身上，所以需要对数据进行标准化，分析结果显示只对方差做标准化、不对均值做标准化比对方差和均值做标准化的效果要好。

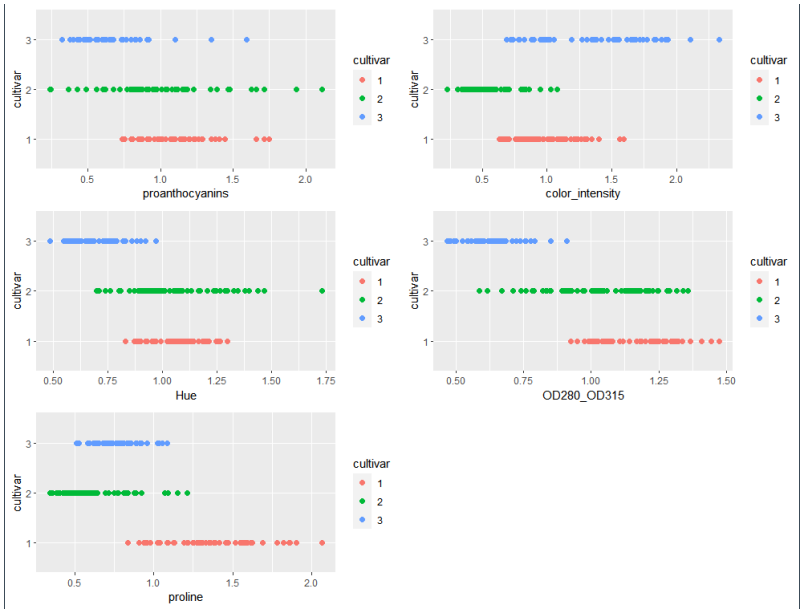
```
 cultivar    alcohol          acid            ash        ash_alkalinity         Mg
 1:59     Min.   :0.8444   Min.   :0.2851   Min.   :0.5693   Min.   :0.5345   Min.   :0.6928
 2:71     1st Qu.:0.9464   1st Qu.:0.6174   1st Qu.:0.9251   1st Qu.:0.8672   1st Qu.:0.8710
 3:48     Median :0.9990   Median :0.7185   Median :0.9879   Median :0.9832   Median :0.9699
          Mean   :0.9953   Mean   :0.9001   Mean   :0.9906   Mean   :0.9829   Mean   :0.9872
          3rd Qu.:1.0471   3rd Qu.:1.1876   3rd Qu.:1.0705   3rd Qu.:1.0840   3rd Qu.:1.0590
          Max.   :1.1353   Max.   :2.2345   Max.   :1.3520   Max.   :1.5126   Max.   :1.6034
    phenols        flavanoids     nonflavanoid    proanthocyanins color_intensity      Hue
 Min.   :0.4109   Min.   :0.1500   Min.   :0.3389   Min.   :0.2419   Min.   :0.2295   Min.   :0.4863
 1st Qu.:0.7306   1st Qu.:0.5316   1st Qu.:0.7038   1st Qu.:0.7375   1st Qu.:0.5774   1st Qu.:0.7928
 Median :0.9874   Median :0.9418   Median :0.8863   Median :0.9174   Median :0.8409   Median :0.9777
 Mean   :0.9622   Mean   :0.8952   Mean   :0.9433   Mean   :0.9386   Mean   :0.9070   Mean   :0.9701
 3rd Qu.:1.1739   3rd Qu.:1.2682   3rd Qu.:1.1404   3rd Qu.:1.1505   3rd Qu.:1.1117   3rd Qu.:1.1348
 Max.   :1.6267   Max.   :2.2409   Max.   :1.7204   Max.   :2.1122   Max.   :2.3310   Max.   :1.7326
  OD280_OD315       proline       cultivar.pred cultivar.cvpred
 Min.   :0.4680   Min.   :0.3422   1:59          1:60
 1st Qu.:0.7140   1st Qu.:0.6160   2:71          2:69
 Median :1.0245   Median :0.8289   3:48          3:49
 Mean   :0.9624   Mean   :0.9192
 3rd Qu.:1.1682   3rd Qu.:1.2123
 Max.   :1.4741   Max.   :2.0677
```

数据标准化后，方差得到了很大的改善，以下分析均使用标准化后的数据。

为单独观察各变量对葡萄酒品种的影响，绘制散点图，横坐标为变换后的单个变量数值，纵坐标为葡萄酒品种1、2、3。由以下13张散点图可知，不同品种葡萄酒的酒精含量（alcohol）、黄酮类物质含量（flavanoids）、稀释后的OD280/OD315比值（OD280_OD315）有较明显的差距，它们可能是区分葡萄酒种类的关键因素；灰分（ash）、镁含量（Mg）、非黄酮类苯酚含量（nonflavanoid phenols）等变量数值分布有明显的重叠现象，难以根据它们对葡萄酒做出分类。总体来看，我们需要多个变量数据对酒的品种做出分类。







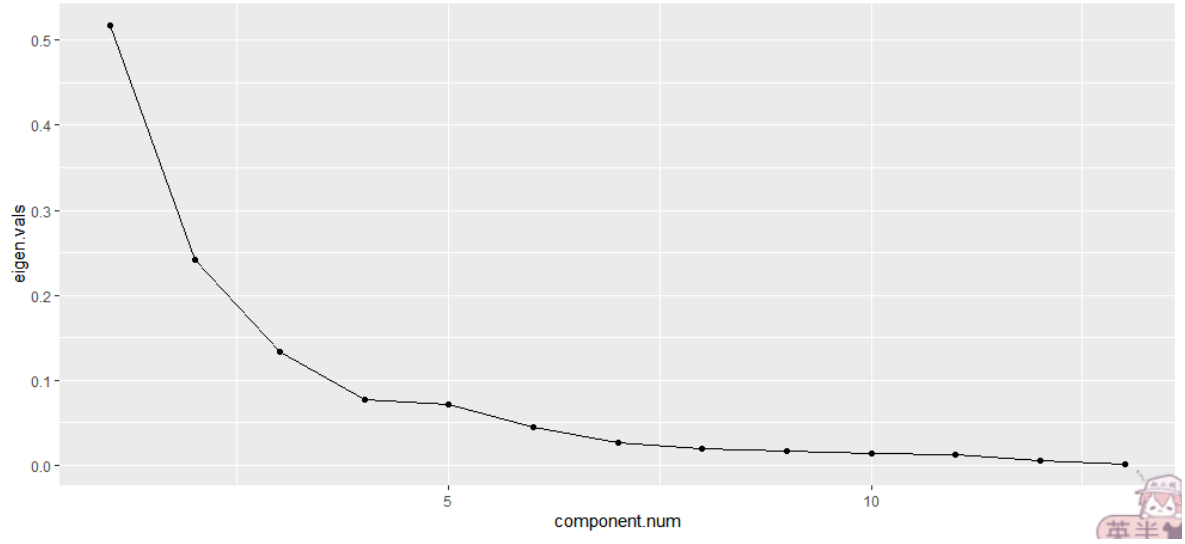
从上面数据中还观察到，品种一的原花青素含量（proanthocyanins），品种二的镁含量（Mg）、酸度（acid），品种三的酸度（acid）等数值明显不服从正态分布，因此有必要对数据进行正态性检验。对3个品种各自13项数值进行正态检验，发现有较多数据不满足正态分布，因此在接下来的分析中应尽量避免正态性假设，若需要正态性假设应先对数据进行进一步处理。

var / pvalue	alcohol	acid	ash	ash_alkalinity	Mg	phenols	flavanoids	nonflavanoid	proanthocyanins	color_intensity	Hue	OD280/OD315	proline
cultivar1	0.479068062	1.20E-10	0.155562576	0.216086353	0.086173806	2.03E-02	0.638726381	3.02E-02	3.15E-02	0.125093539	0.150821545	0.07744586	0.523240698
cultivar2	0.113960721	1.84E-07	0.619761444	0.073973835	5.79E-09	0.318011865	1.50E-03	0.312804121	8.15E-03	8.19E-04	0.224929555	0.089039583	1.77E-03
cultivar3	0.640838702	0.737718233	0.109227279	0.098742177	3.87E-02	1.58E-02	3.56E-04	2.28E-02	2.49E-04	0.087752319	2.82E-02	0.083107149	0.458494504

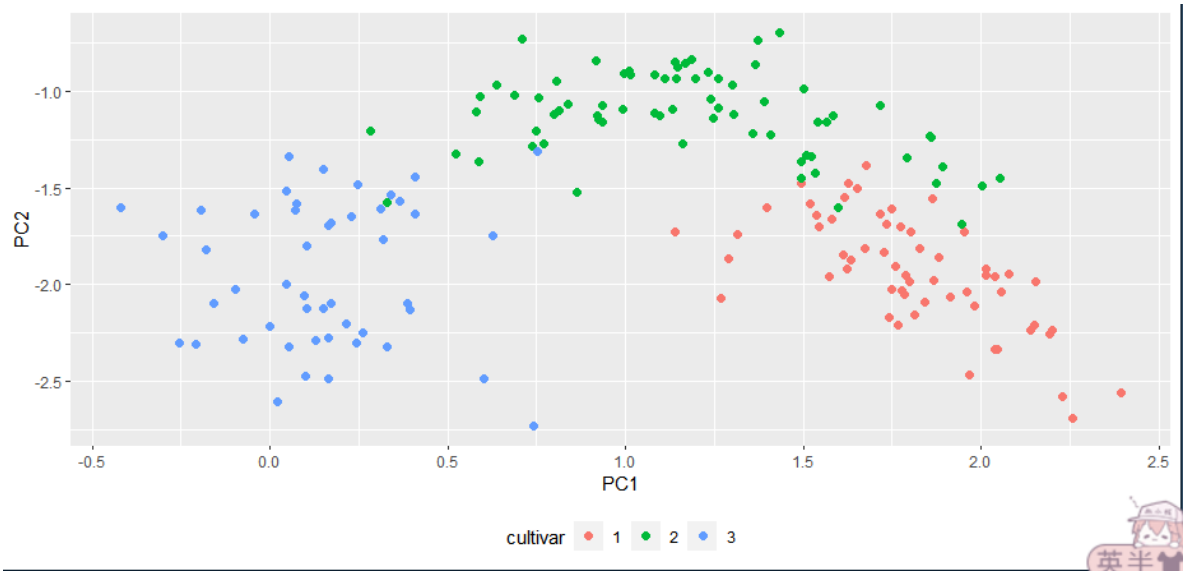
三、建模

3.1主成分分析

对数据进行主成分分析（PCA），画出崖底碎石图：



前五个主成分已经可以解释数据88.3%的方差，因此在后续判别分析中可以将前五个主成分作为输入变量。将前两项主成分作为横轴和纵轴，绘制如下散点图。可以发现，三个品种的葡萄酒基本被分开，因此在接下来观察分类效果时可以将前两个主成分作为横纵轴。



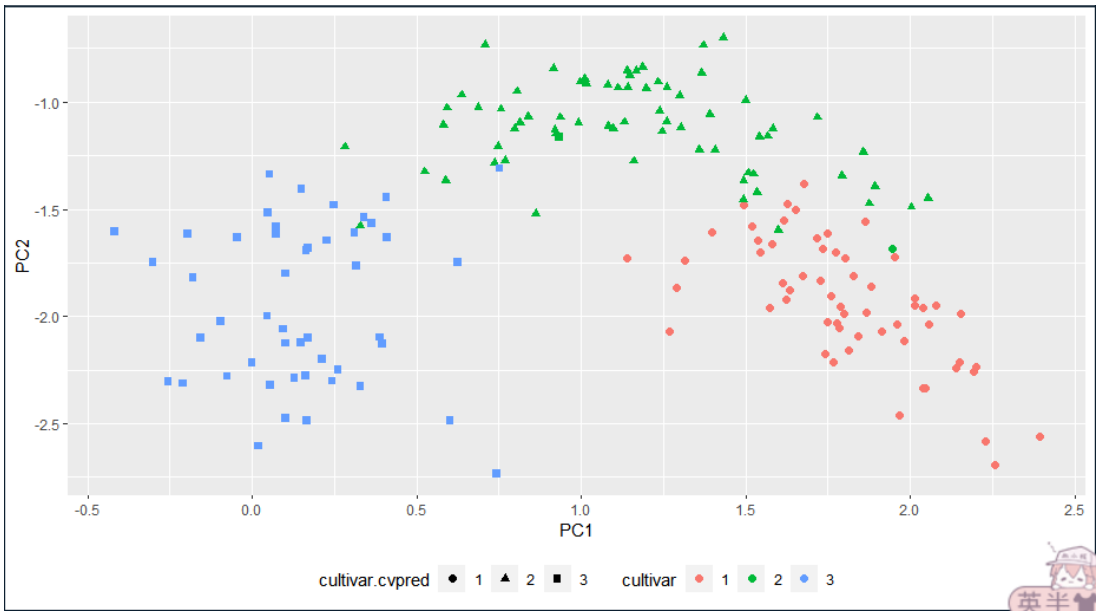
3.2因子分析

我们知道，酚类物质的含量会影响样品在280nm处的吸收峰（OD280），脯氨酸含量对蛋白质的吸收峰有影响，灰分和灰分碱度也有联系，因此这13个变量背后存在共有因子影响，对数据进行因子分析，当取因子数目为5时，每个变量的独有特性均较少，其大部分特征都被5个因子解释了，这与主成分分析时得到的结果相吻合。以下5个因子解释了数据的80%的变异性。

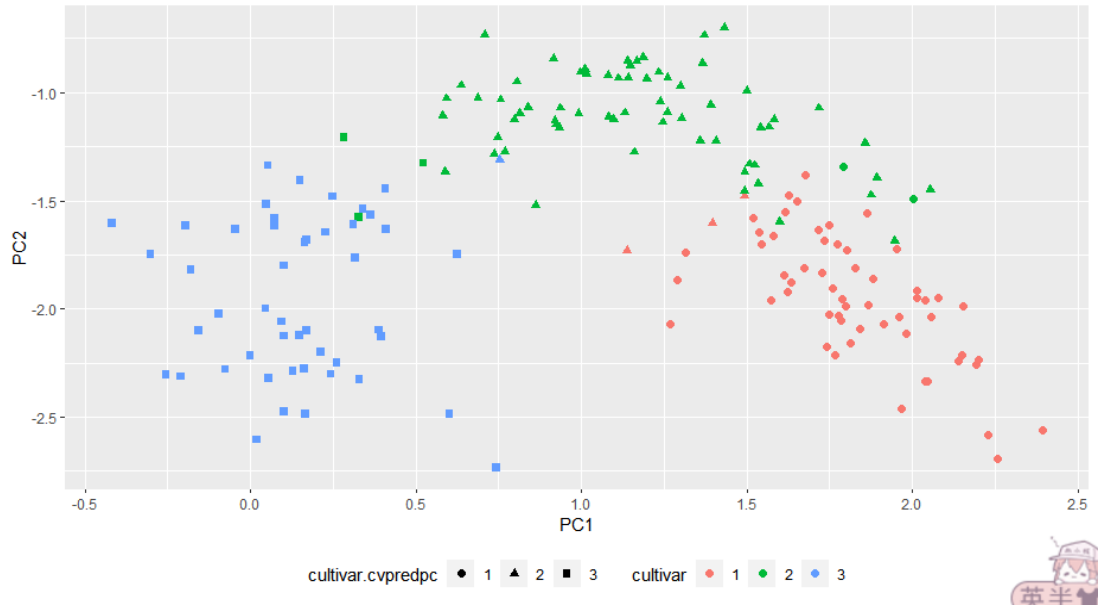
```
Principal Components Analysis
Call: principal(r = data[, 2:n], nfactors = 5, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                   RC1   RC2   RC4   RC3   RC5   h2    u2 com
alcohol           0.15  0.88 -0.11 -0.01  0.06 0.80 0.195 1.1
acid             -0.15  0.01 -0.79  0.17 -0.11 0.69 0.313 1.2
ash               0.08  0.24 -0.03  0.89  0.16 0.88 0.124 1.2
ash_alkalinity   -0.18 -0.42 -0.29  0.72 -0.01 0.82 0.181 2.2
Mg                0.10  0.23  0.05  0.15  0.91 0.91 0.091 1.2
phenols           0.84  0.27  0.24  0.03  0.03 0.83 0.170 1.4
flavanoids        0.87  0.19  0.34  0.00  0.05 0.91 0.094 1.4
nonflavanoid     -0.57  0.00 -0.07  0.44 -0.44 0.72 0.285 2.9
proanthocyanins   0.79  0.05 -0.03 -0.04  0.15 0.66 0.339 1.1
color_intensity  -0.20  0.70 -0.47  0.12  0.12 0.77 0.225 2.1
Hue               0.35 -0.09  0.83 -0.02 -0.02 0.81 0.187 1.4
OD280_OD315       0.82 -0.06  0.37 -0.05 -0.05 0.81 0.187 1.4
proline           0.32  0.78  0.25 -0.01  0.22 0.81 0.187 1.8
```

3.3判别分析

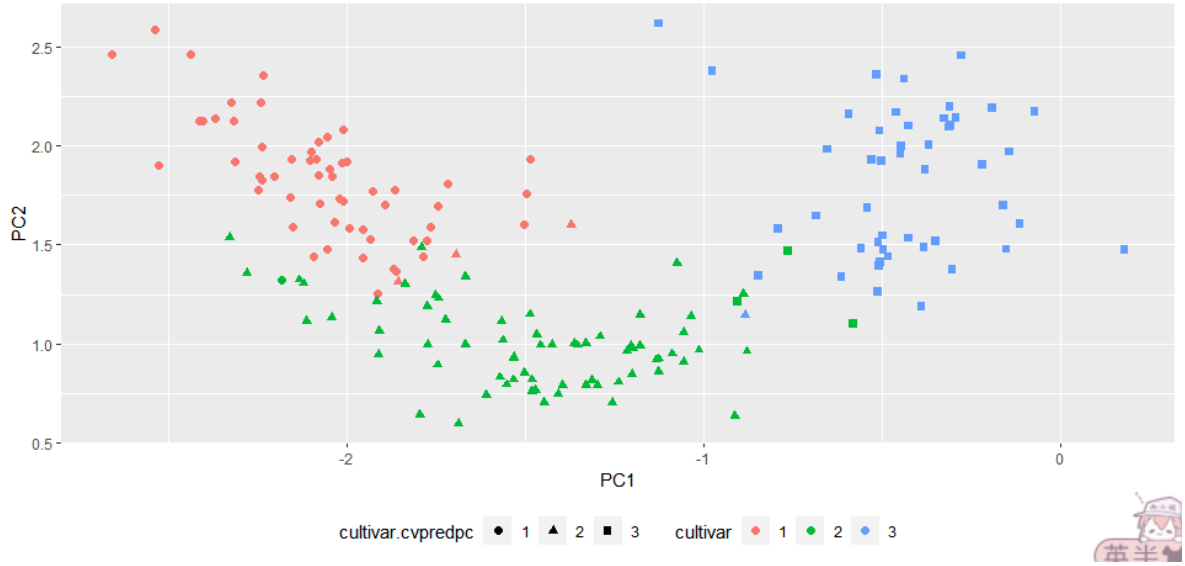
利用线性判别分析对数据进行分类，首先将所有的13个变量作为输入，对数据进行交叉验证，错误率为1.12%。从图中可以发现，有1个品种二的葡萄酒被误分为了品种一，1个品种二的葡萄酒杯误分为了品种3。



上述主成分分析得出结论，前五个主成分已经对数据解释了88.3%的变异性，因此考虑将这5个主成分作为输入进行线性判别分析。对数据进行交叉验证，错误率为5.06%。有9个数据点被误判，虽然错误率比将原始数据作为输入时高，但是经过降维处理后计算量大大减小，且失误率在可接受范围内。



在对数据做探索性分析时发现，灰分（ash）、镁含量（Mg）、非黄酮类苯酚含量（nonflavanoid phenols）等变量数值分布重叠严重，似乎对分类没有贡献，因此可以大胆尝试将这3个变量从数据集中去掉，重新做主成分分析。这时发现，仅前4个主成分对数据方差的解释度就达到了89%。取这4个主成分作为输入进行线性判别分析，对数据进行交叉验证，结果显示错误率降低到了4.49%。也就是说，去掉预处理时发现的疑似无用变量后，不仅降低了计算的复杂度，同时还提高了分类的正确率！



四、结论及改进

数据预处理对本次研究非常重要，通过对数据的探索性分析，决定对原始数据的方差做归一化，去掉重叠程度大的变量，对处理后的数据进行主成分分析和因子分析。主成分分析给出了解释性很高的结果，它表明前4个主成分就可以解释数据大部分的变异性。因此将这4个主成分作为输入进行线性判别分析，最终得到了在此数据集上正确率95.1%的分类模型。

本实验的不足之处在于线性分类器对于非线性的部分是无法解释的，采用Kernel Fisher Discriminant等非线性分类方法可能会得到更好的结果。此外，在预处理部分我通过直接观察扔掉了3个变量，这个处理还是有些粗糙的。可以对三个品种的同一变量之间进行线性回归分析来判断其是否存在多重共线性，这样可以得到更精确的处理。

五、附录：源代码

```
library(ggplot2)
library(ggpubr)
library(psych)
library(MASS)

##################  read data  ###################
data = read.table('wine.dat', sep = ',')

names(data) = c('cultivar', 'alcohol', 'acid',
                    'ash', 'ash_alkalinity', 'Mg',
                    'phenols', 'flavanoids', 'nonflavanoid',
                    'proanthocyanins', 'color_intensity', 'Hue',
                    'OD280_OD315', 'proline')

data$cultivar = factor(data$cultivar)

summary(data)

n = length(data)

#normalize
#note that data should not be centered
data[, 2:n] = scale(data[, 2:n], center = FALSE, scale = TRUE)
###################################################


#############  observe each variable  ############
p1 = ggplot(data, aes(x=alcohol, y=cultivar, color=cultivar)) + geom_point(size=2)
p2 = ggplot(data, aes(x=acid, y=cultivar, color=cultivar)) + geom_point(size=2)
p3 = ggplot(data, aes(x=ash, y=cultivar, color=cultivar)) + geom_point(size=2)
p4 = ggplot(data, aes(x=ash_alkalinity, y=cultivar, color=cultivar)) + geom_point(size=2)
ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)
```

```r
p5 = ggplot(data, aes(x=Mg, y=cultivar, color=cultivar)) + geom_point(size=2)
p6 = ggplot(data, aes(x=phenols, y=cultivar, color=cultivar)) + geom_point(size=2)
p7 = ggplot(data, aes(x=flavanoids, y=cultivar, color=cultivar)) + geom_point(size=2)
p8 = ggplot(data, aes(x=nonflavanoid, y=cultivar, color=cultivar)) + geom_point(size=2)
ggarrange(p5, p6, p7, p8, ncol = 2, nrow = 2)
p9 = ggplot(data, aes(x=proanthocyanins, y=cultivar, color=cultivar)) + geom_point(size=2)
p10 = ggplot(data, aes(x=color_intensity, y=cultivar, color=cultivar)) + geom_point(size=2)
p11 = ggplot(data, aes(x=Hue, y=cultivar, color=cultivar)) + geom_point(size=2)
p12 = ggplot(data, aes(x=OD280_OD315, y=cultivar, color=cultivar)) +geom_point(size=2)
p13 = ggplot(data, aes(x=proline, y=cultivar, color=cultivar)) +geom_point(size=2)
ggarrange(p9, p10, p11, p12, p13, ncol = 2, nrow = 3)
#####################################################


################  Normality test  ################
type1 = data[data$cultivar == 1, ]
type2 = data[data$cultivar == 2, ]
type3 = data[data$cultivar == 3, ]
shapiro_pvalue = array(0, dim = c(3, n-1))
for (i in 1:(n-1)) {
  shapiro_pvalue[1, i] = shapiro.test(type1[, i+1])$p.value
}
for (i in 1:(n-1)) {
  shapiro_pvalue[2, i] = shapiro.test(type2[, i+1])$p.value
}
for (i in 1:(n-1)) {
  shapiro_pvalue[3, i] = shapiro.test(type3[, i+1])$p.value
}
rownames(shapiro_pvalue) = c('cultivar1', 'cultivar2', 'cultivar3')
colnames(shapiro_pvalue) = c('alcohol', 'acid', 'ash', 'ash_alkalinity',
                             'Mg', 'phenols', 'flavanoids', 'nonflavanoid',
                             'proanthocyanins', 'color_intensity', 'Hue',
                             'OD280/OD315', 'proline')
write.csv(shapiro_pvalue, 'shapiro test(scaled).csv')
#####################################################


####################  PCA  ####################
winepca <- prcomp(data[,2:n])
evals <- data.frame(winepca$sdev^2)
#winepca$sdev: return the standard deviations of the principal components
names(evals) <-'eigen.vals'
evals$component.num <- as.integer(seq(nrow(evals)))
ggplot(evals,aes(x=component.num,y=eigen.vals)) +
  geom_point() +
  geom_line()
Gamma <- winepca$rot
#winepca$rot: return the matrix of variable loadings
round(Gamma,4)
summary(winepca)
#####################################################


####################  FA  ####################
#PC method
#shapiro test tells me that nomality is not always guaranteed
winefa = principal(data[, 2:n], nfactors=5, rotate='varimax')
winefa # print results

# plot(winefa$values,type='b') # scree plot
# plot(winefa$loadings)
# plot(winefa$loadings,type="n") # set up plot
# text(winefa$loadings,labels=names(data[, 2:n]),cex=.7) # add variable names
#####################################################

####################  DA  ####################

#LDA for pcs
pcs = as.matrix(data[, 2:n])%*%as.matrix(Gamma[, 1:5])
pcdata = data.frame(data$cultivar, pcs)
names(pcdata) = c('cultivar', 'PC1', 'PC2', 'PC3', 'PC4', 'PC5')
ggplot(data=pcdata, aes(x=PC1, y=PC2, col=cultivar)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')
#Accuracy of LDA using cross validation
cvLpc = lda(cultivar ~ PC1 + PC2 + PC3 + PC4 + PC5, data=pcdata, CV=TRUE)
pcdata$cultivar.cvpredpc = cvLpc$class
ggplot(pcdata,aes(x=PC1,y=PC2,col=cultivar, shape=cultivar.cvpredpc)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')
tabcvpc = table(pred=pcdata$cultivar.cvpredpc, true=pcdata$cultivar);tabcvpc
cverrpc = sum(tabcvpc[row(tabcvpc)!=col(tabcvpc)])/sum(tabcvpc);cverrpc

#Predicting labels using LDA with original data.
L = lda(cultivar ~ data$alcohol + data$acid + data$ash +
```

```
          data$ash_alkalinity + data$Mg + data$phenols +
          data$flavanoids + data$nonflavanoid + data$proanthocyanins +
          data$color_intensity + data$Hue + data$OD280_OD315 +
          data$proline, data=data)
data$cultivar.pred = predict(L, data)$class
ggplot(data,aes(x=pcs[, 1],y=pcs[, 2],col=cultivar, shape=cultivar.pred)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')

#Accuracy of LDA
tab = table(pred=data$cultivar.pred, true=data$cultivar);tab
aper = sum(tab[row(tab)!=col(tab)])/sum(tab);aper

#Accuracy of LDA using cross validation
cvL = lda(cultivar ~ data$alcohol + data$acid + data$ash +
          data$ash_alkalinity + data$Mg + data$phenols +
          data$flavanoids + data$nonflavanoid + data$proanthocyanins +
          data$color_intensity + data$Hue + data$OD280_OD315 +
          data$proline, data=data, CV=TRUE)
data$cultivar.cvpred = cvL$class
ggplot(data,aes(x=pcs[, 1],y=pcs[, 2],col=cultivar, shape=cultivar.cvpred)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')
tabcv = table(pred=data$cultivar.cvpred, true=data$cultivar);tabcv
cverr = sum(tabcv[row(tabcv)!=col(tabcv)])/sum(tabcv);cverr
##################################################


############  throw several variables  ############
reduced_data = data.frame(data$cultivar, data$alcohol, data$acid,
                          data$ash_alkalinity,data$phenols,
                          data$flavanoids, data$proanthocyanins,
                          data$color_intensity, data$Hue,
                          data$OD280_OD315, data$proline)
names(reduced_data) = c('cultivar', 'alcohol', 'acid',
                        'ash_alkalinity', 'phenols', 'flavanoids',
                        'proanthocyanins', 'color_intensity', 'Hue',
                        'OD280_OD315', 'proline')
reduced_pca <- prcomp(reduced_data[,2:11])
reduced_evals <- data.frame(reduced_pca$sdev^2)
#reduced_pca$sdev: return the standard deviations of the principal components
names(reduced_evals) <-'eigen.vals'
reduced_evals$component.num <- as.integer(seq(nrow(reduced_evals)))
ggplot(reduced_evals,aes(x=component.num,y=eigen.vals)) +
  geom_point() +
  geom_line()
reduced_Gamma <- reduced_pca$rot
#reduced_pca$rot: return the matrix of variable loadings
round(reduced_Gamma,4)
summary(reduced_pca)

#LDA
pcs = as.matrix(reduced_data[, 2:11])%*%as.matrix(reduced_Gamma[, 1:4])
pcdata = data.frame(data$cultivar, pcs)
names(pcdata) = c('cultivar', 'PC1', 'PC2', 'PC3', 'PC4')
ggplot(data=pcdata, aes(x=PC1, y=PC2, col=cultivar)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')
#Accuracy of LDA using cross validation
cvLpc = lda(cultivar ~ PC1 + PC2 + PC3 + PC4, data=pcdata, CV=TRUE)
pcdata$cultivar.cvpredpc = cvLpc$class
ggplot(pcdata,aes(x=PC1,y=PC2,col=cultivar, shape=cultivar.cvpredpc)) +
  geom_point(size=2) +
  xlab('PC1') + ylab('PC2') +
  theme(legend.position='bottom')
tabcvpc = table(pred=pcdata$cultivar.cvpredpc, true=pcdata$cultivar);tabcvpc
cverrpc = sum(tabcvpc[row(tabcvpc)!=col(tabcvpc)])/sum(tabcvpc);cverrpc

#this is great
#I finally choose this model:)
##################################################
```