**Group 9**

Yucheng Lu (G20257365)

Zefeng Song (G22237721)

Linge Yan(G45235107)

Bo Yuan（G30021277）

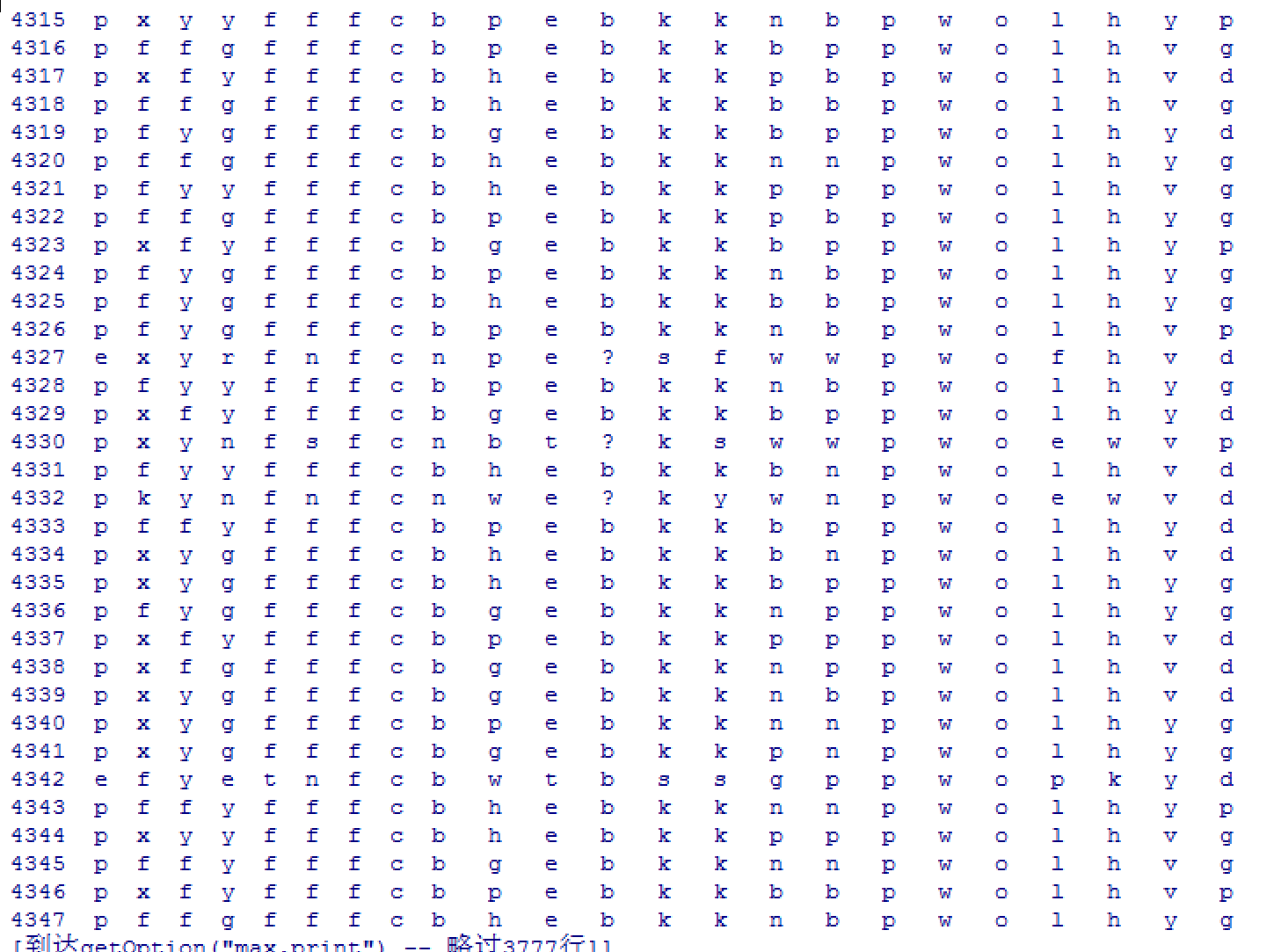
Abdulmohsen Almalki (G21772155)

**Read Data into R**

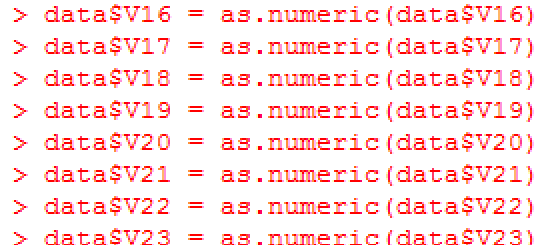
data = read.table("agaricus-lepiota.data", sep = ',')

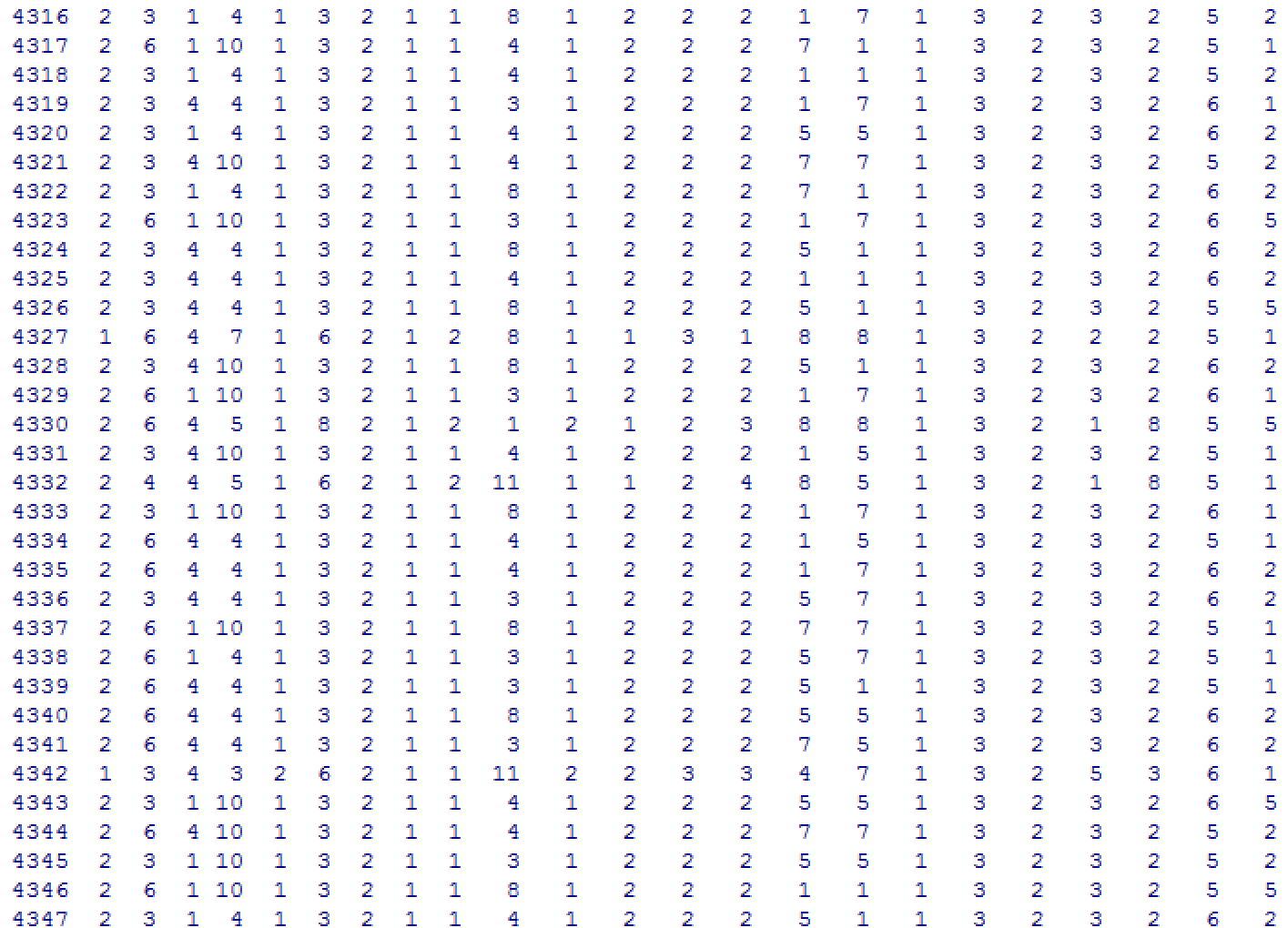
**Convert factor data into numeric data**

Because the original data is factor rather than numeric and thus cannot be used in Principal Components Analysis



data$V1 = as.numeric(data$V1)





we do not need all the attributes and thus we need to remove some attributes with mass missing value. In this dataset, we remove V17.

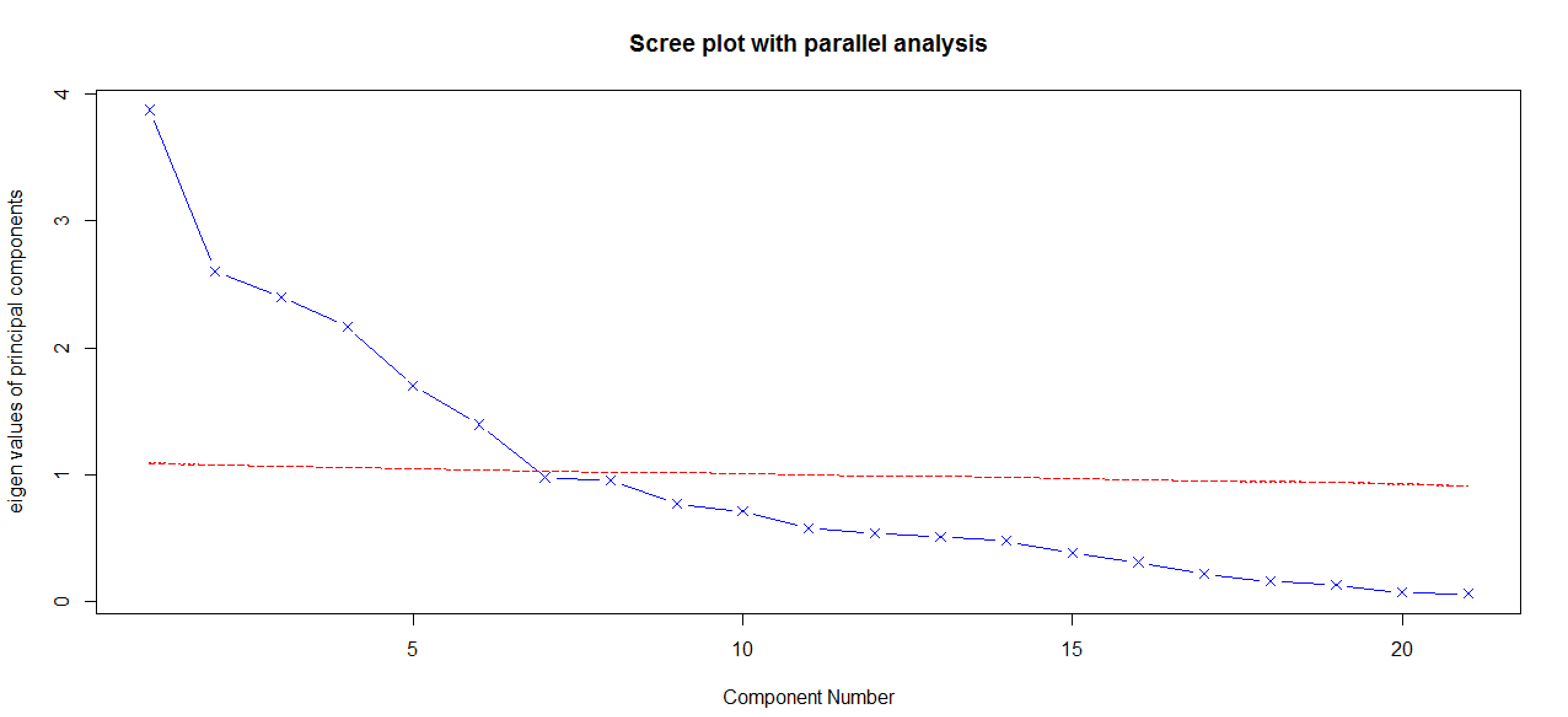
data = data[,-17]

**Principal Components Analysis**

First of all, we need to judge the number of principal components

fa.parallel(data[,-1], fa = 'pc', n.iter = 100,show.legend = F, main = "Scree plot with parallel analysis")

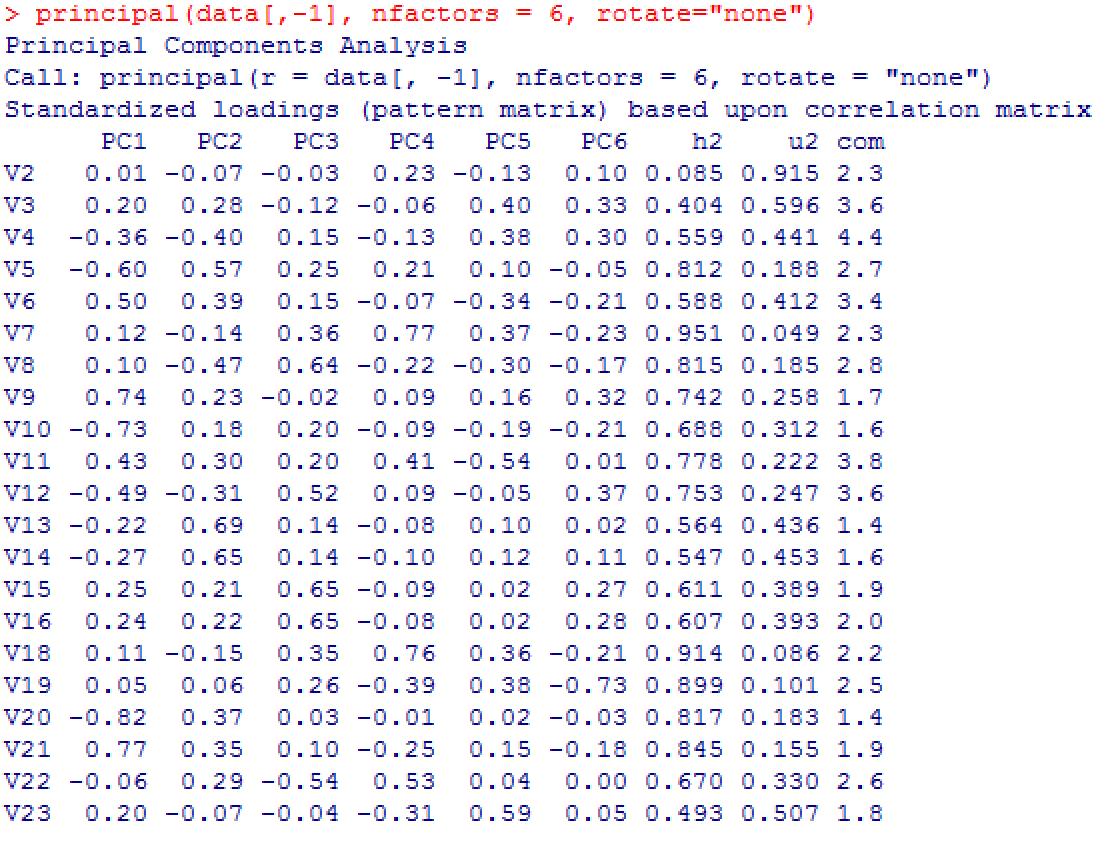


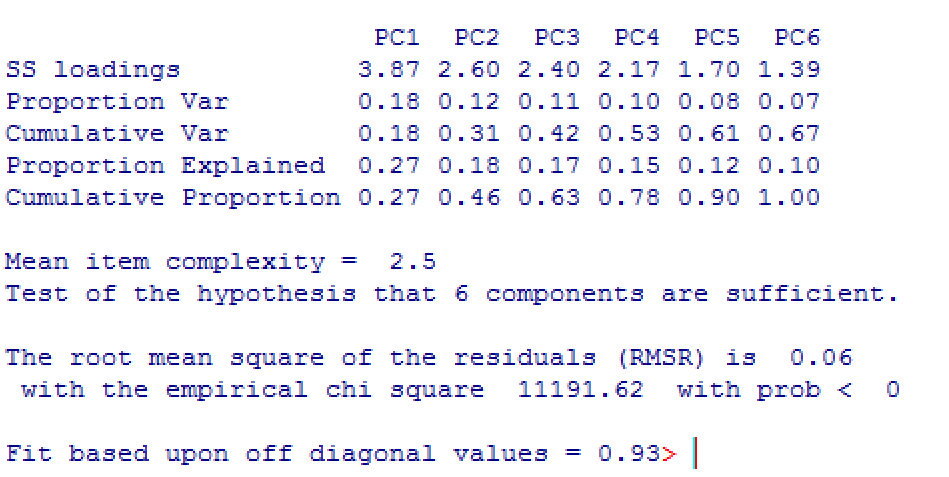


This function to show the scree plot with parallel analysis. The 1 is the standard feature value and the number below this value is what we do not need. Parallel analysis suggests that the number of components = 6.

The next part is to show the result of Principal Components Analysis

principal(data[,-1], nfactors = 6, rotate="none")





From the proportion var and cumulative var. The six principle components’ explanation rate is 67%.

We may can use these 6 new attributes to explain whole model.

3.

4.

1. **lm() function**

R linear regression uses the lm() function to create a regression model given some formula, in the form of Y~X+X2.

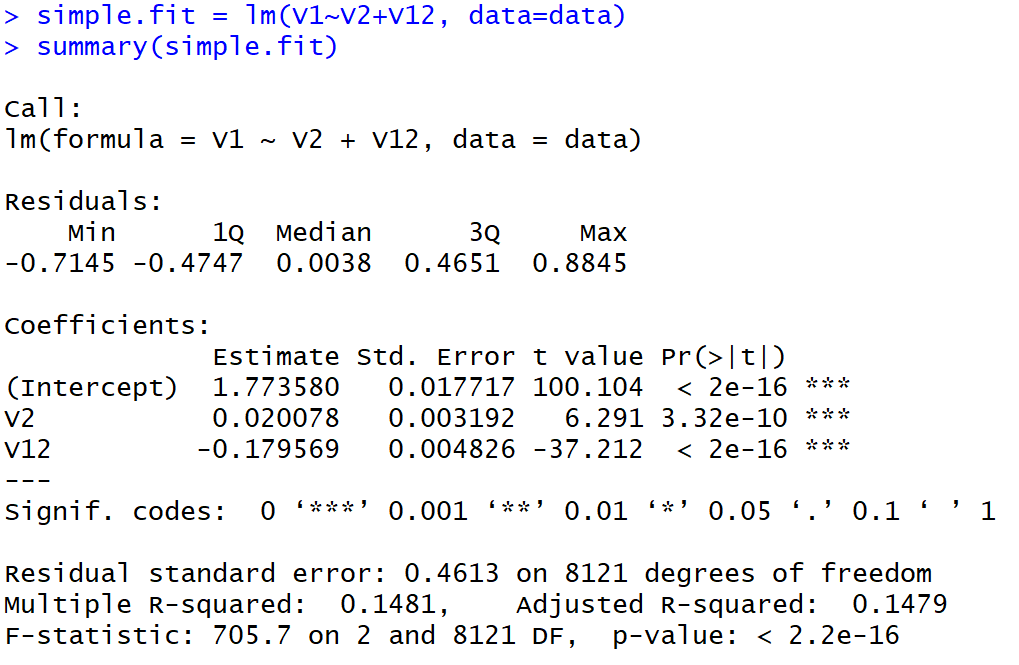
To do linear (simple and multiple) regression in R, we need the built-in lm function.

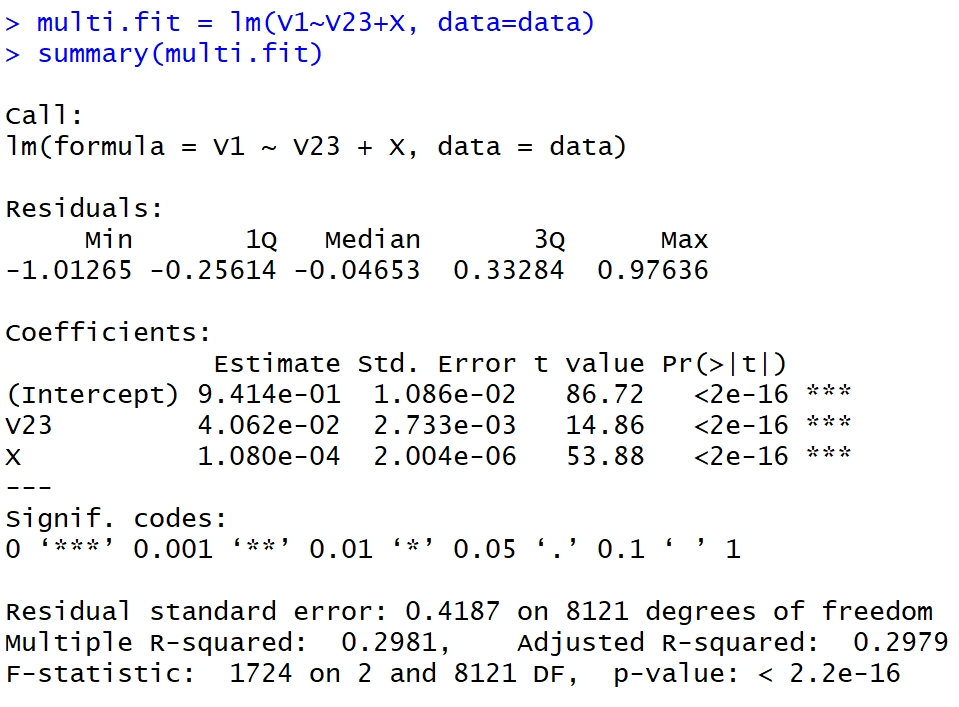
lm(formula, data, subset, weights, na.action,

method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,

singular.ok = TRUE, contrasts = NULL, offset, …)

To look at the model, we use the summary() function.





2. **glm() function**

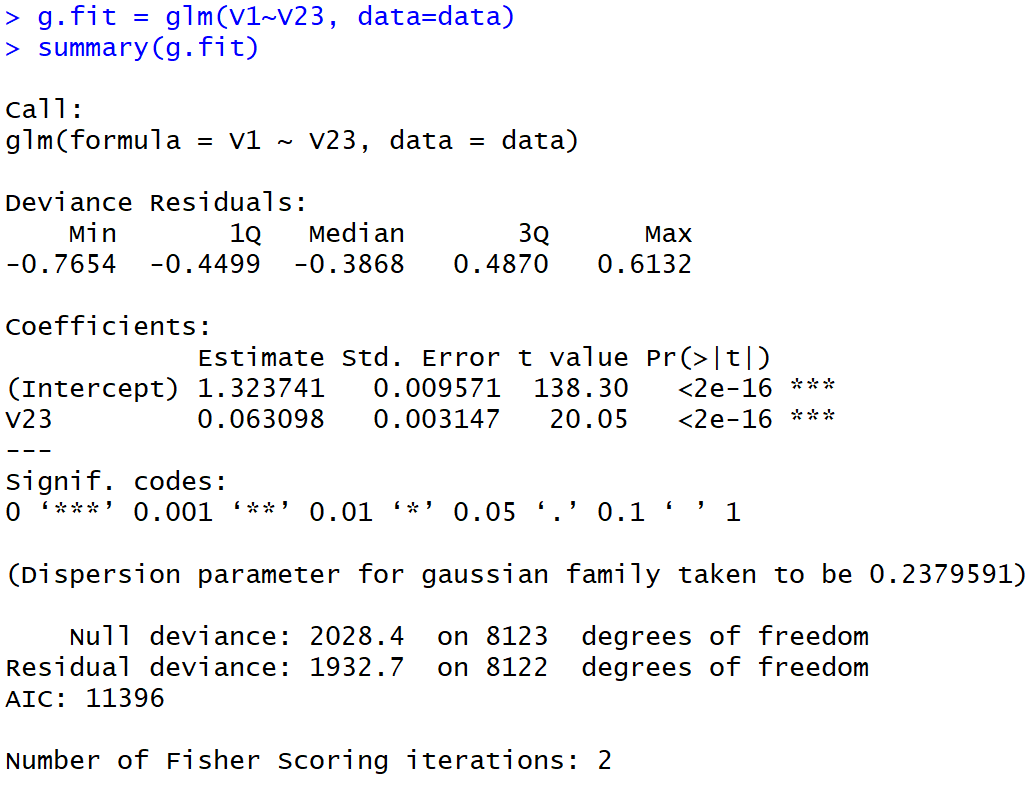
glm is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.

glm(formula, family = gaussian, data, weights, subset,

na.action, start = NULL, etastart, mustart, offset,

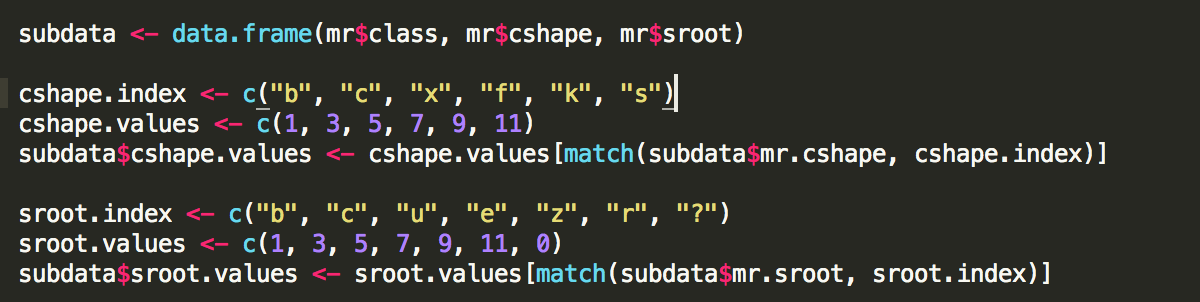
control = list(…), model = TRUE, method = "glm.fit",

x = FALSE, y = TRUE, contrasts = NULL, …)



5. using SVM

loading the data, selecting some effective attributes and giving them the values from “1”, “3”, “5” …



I select the cshape and sroot and set the values for each index:

Setting values:

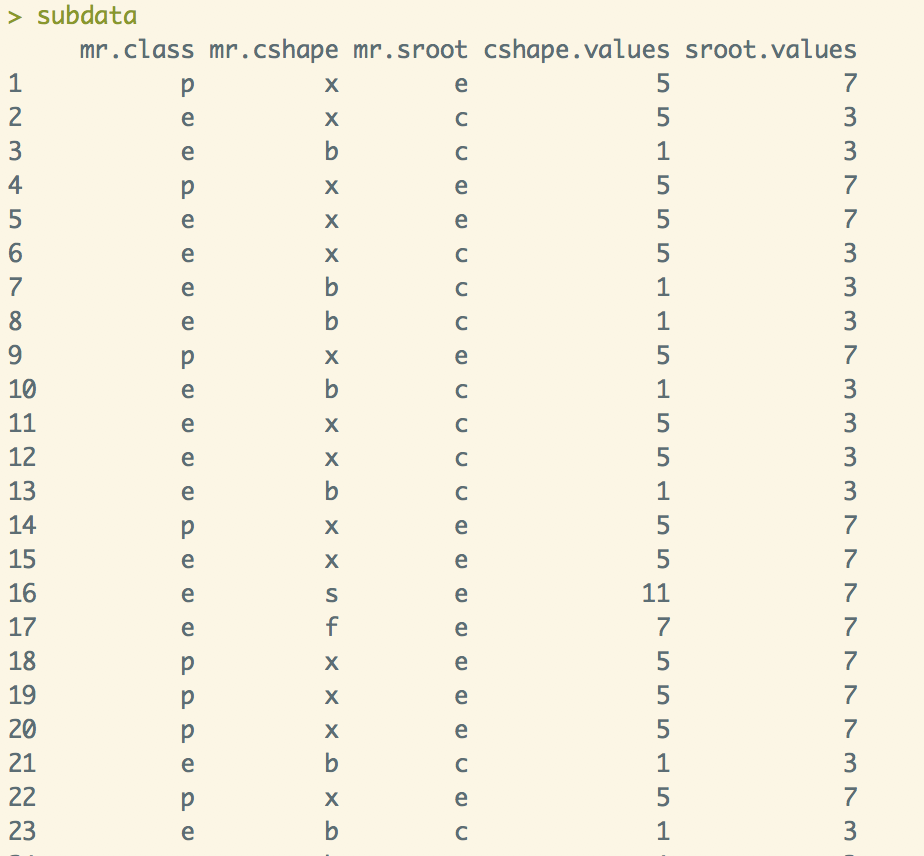
1. cap-shape: bell=b, conical=c, convex=x, lat=f, knobbed=k, sunken=s

b =1, c =3, x = 5, f = 7, k=9, s =11

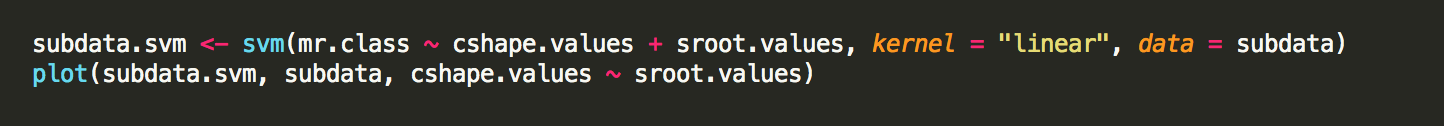
2.stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?

b=1, c=3, u =5, e= 7, z=9, r=11, ?= 0

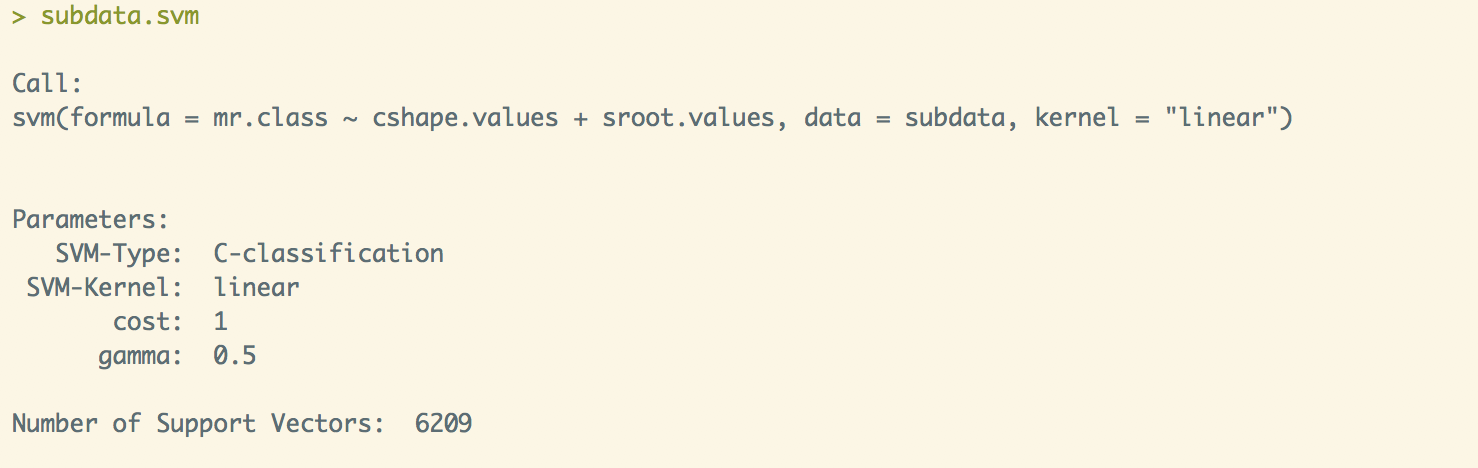
check the table:

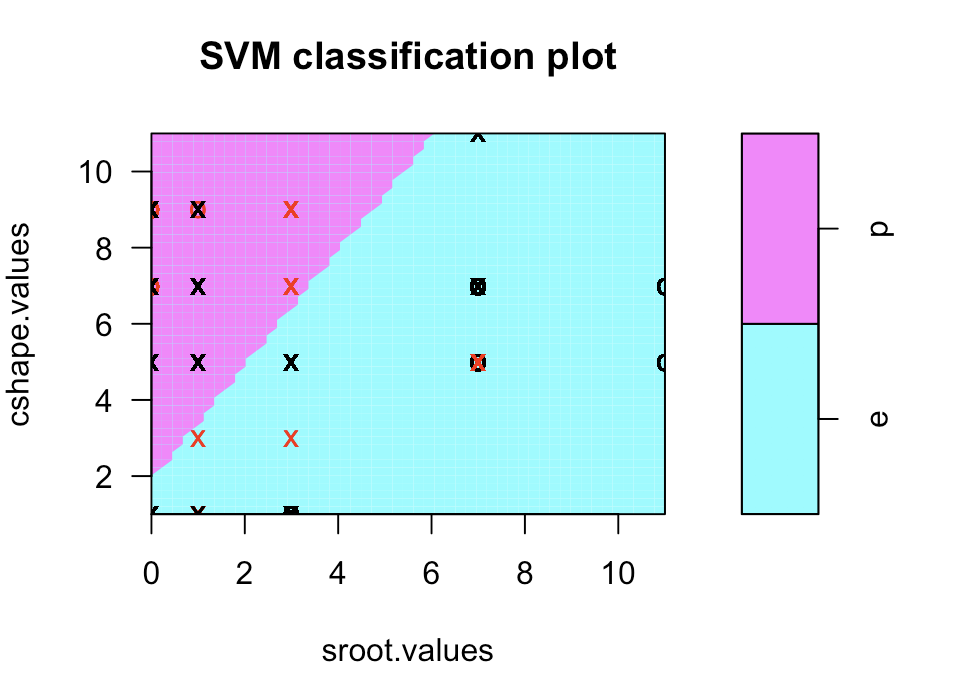


Using the SVM function:



see the result and plot:





What We Know:

We can see the this picture, p class mushroom has a high values of Cshape and low values of Sroot: e class mushroom has a high values of Sroot and low values of Cshape. That means p class mushroom likely to has sunken, knobbed and bulbous: The e mushroom likely has conical, bell and rooted.

6.e

The SVM help us to understand the data best as it can classify the class of the mushroom best. From the Picture of SVM(Using the cap-shape and stalk-root attributes to calculate), we can easily see the linear relationship. p class mushroom has a high values of Cshape(flat, knobbed and sunken) and low values of Sroot(bulbous, club and cup): e class mushroom has a high values of Sroot(equal, rhizomorphs and rooted) and low values of Cshape(bell, conical and convex).

What we learn

From the SVM part of the project, we know how the SVM functions work, how to use it to analysis the data and how the work with other teammates when facing the difficulty. SVM is a very useful method for big data, we still need to learn more to master it.

6.f

From this project we can understand that in Data science we may not be one hundred percent sure about the final result. However, we can get an estimate about the data. For example, we used k-means, k-medoids, k-nearest neighbor algorithms for clustering, and we have got different results for different clusters.