Mid-term Exam

Huilong An UNI: ha2399

Problem 1:

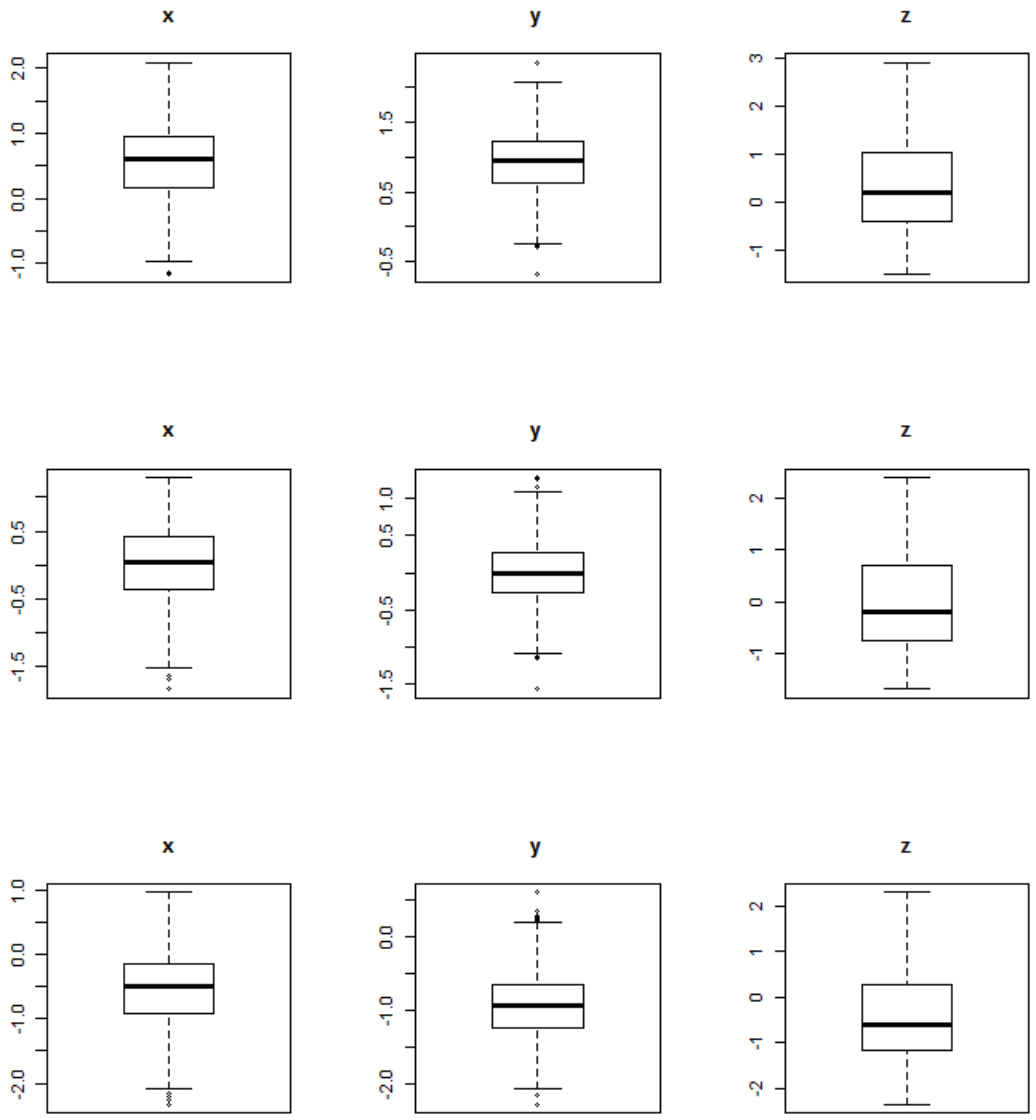
(1)

```
str(data1)# it's a "list"
 num [1:5400, 1:4] 1.412 0.952 1.362 1.733 0.704 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:4] "x" "y" "z" "group"
> attributes(data1)# the attributes of this list
$dim
[1] 5400    4

$dimnames
$dimnames[[1]]
NULL

$dimnames[[2]]
[1] "x"     "y"     "z"     "group"
```

(2)

Before starting, we'd better take a look at the crude distributions of every variable in every group.

The boxplots are shown below:

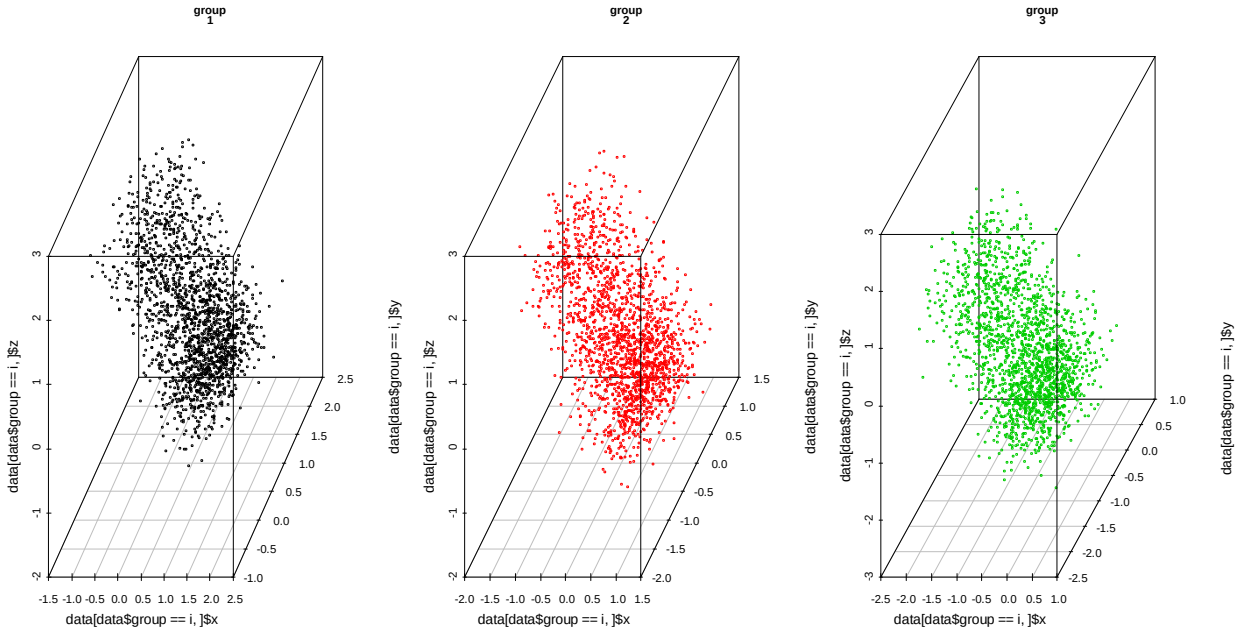


We can find from the boxplot that x and y 's distribution are very plain. And there might exist some extreme values in y in every groups. (outliers alert!)

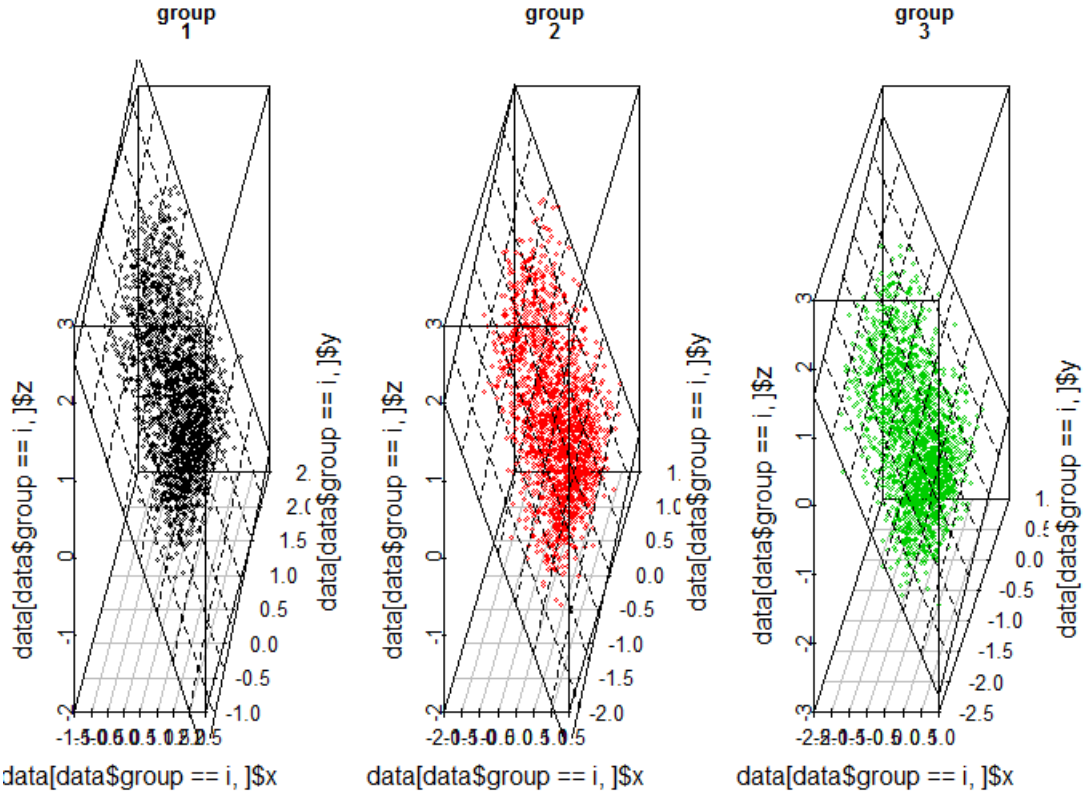
Here, we plot 3-d scatterplots of each group first to see their structure in space respectively. And then put them in one plot to see their relative structures in the space to find some information and get some intuitions for our further analysis:

(1) Are three groups well-separated? Or how complex they are mixed?

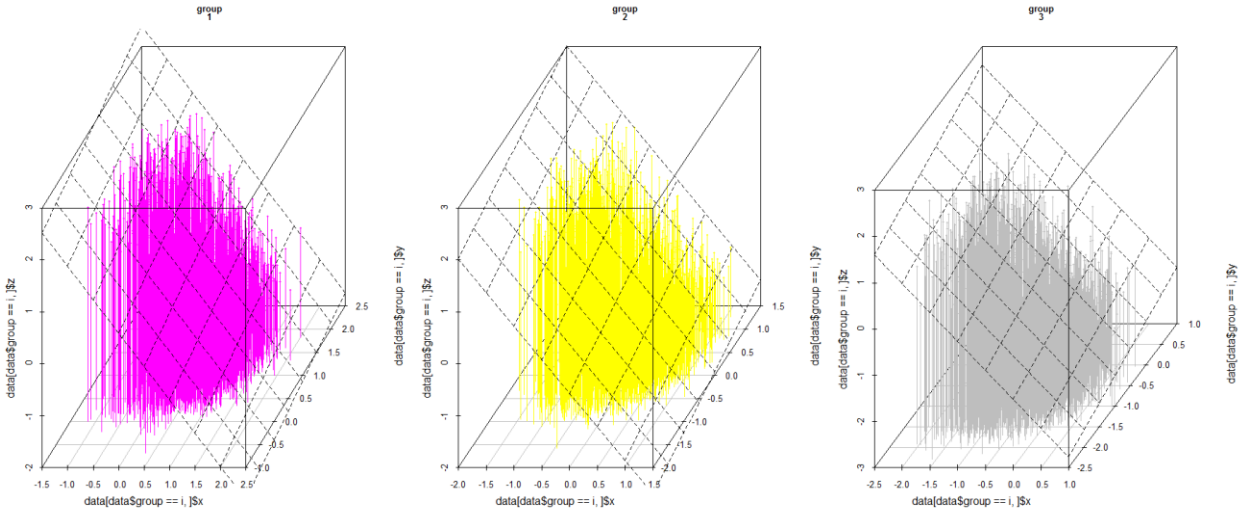
(2 Are they centralized or dispersed?



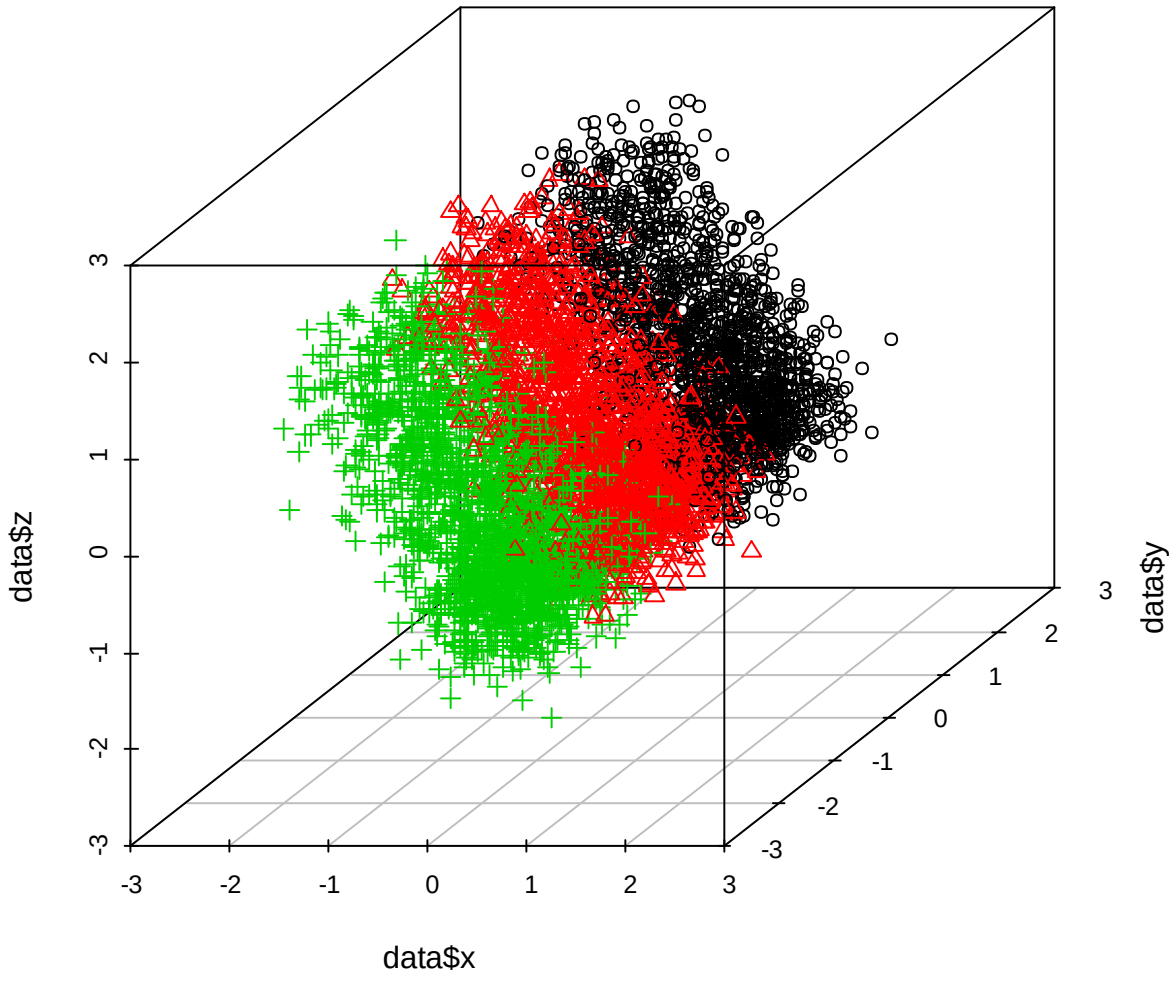
We can find from the respective scatterplots that each group has the similar structure in the space. And it seems that we can put a plane on them.



It is not clear to see how well the points lie in the plane. We can have the plots shown below:

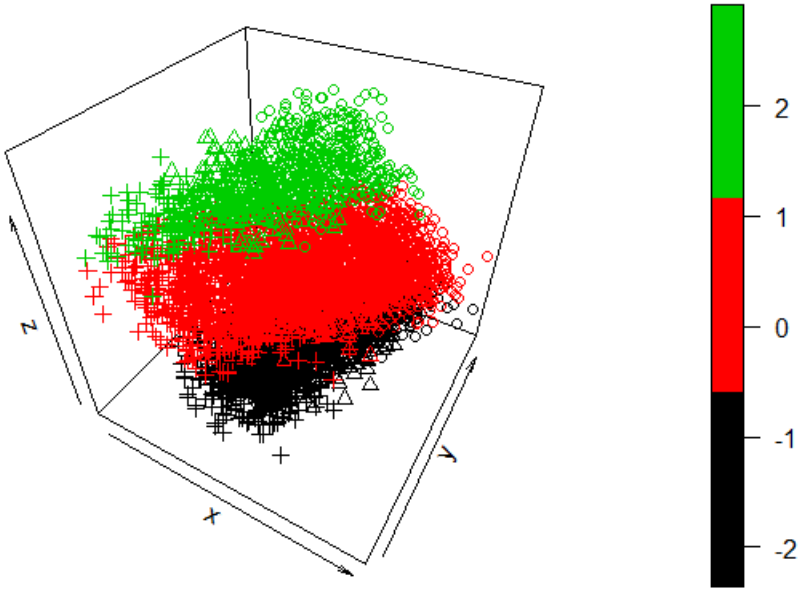


In this plot, we can get the result: not so good, the plane does not a good fit to these points.

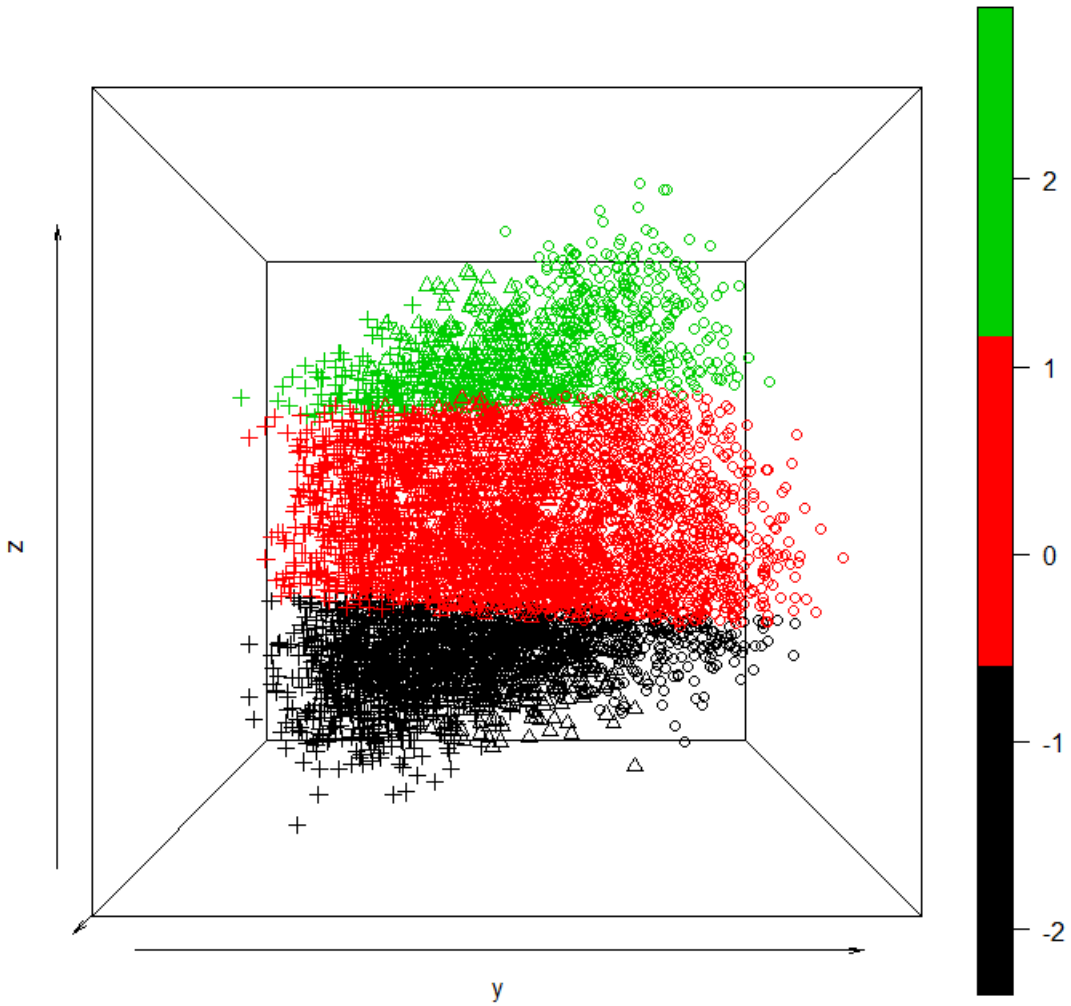


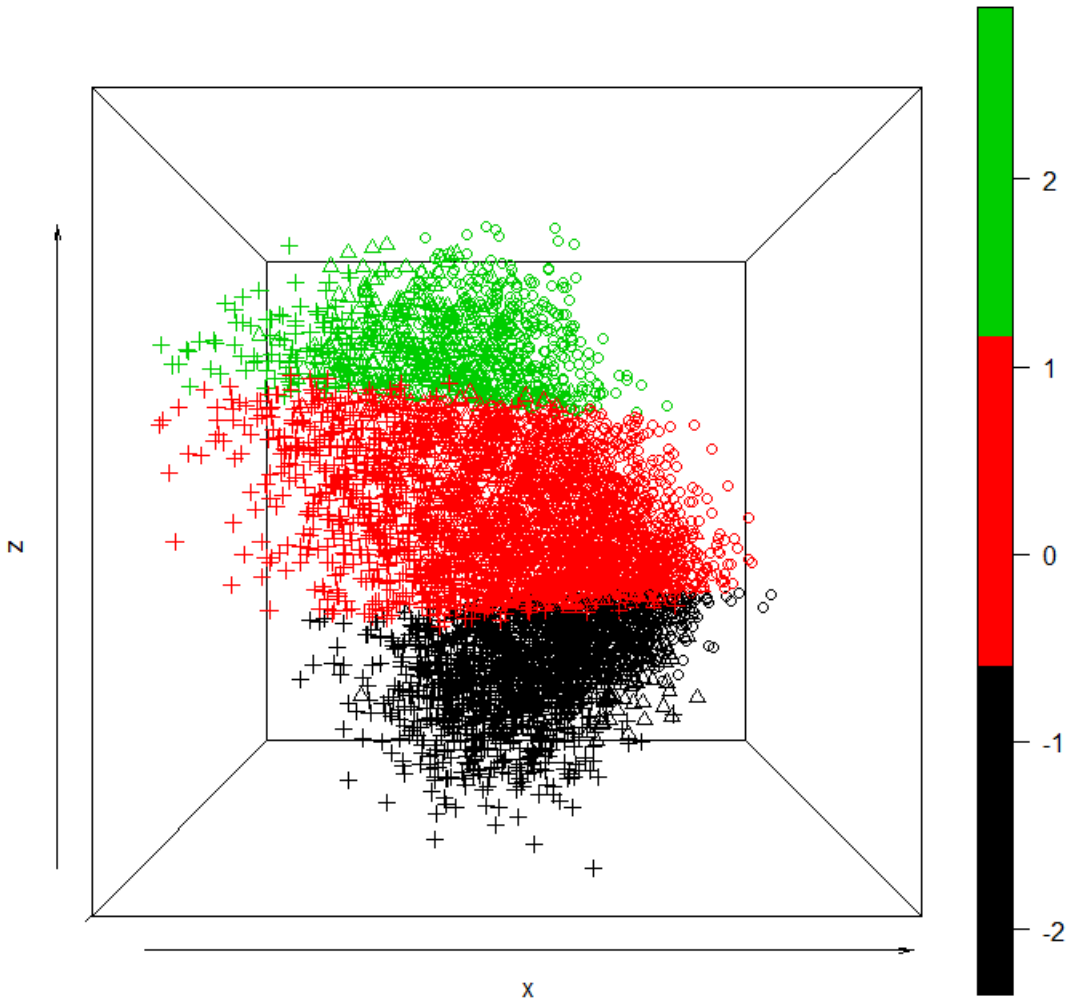
These three groups in the 3-d space seems have some noticeable characters. If we can rotate the plot, we can find more about how these three groups are arranged in the space: if they are well separated or highly-mixed (we can roughly conclude these three groups are not highly mixed because they are separated in the above plot in some sense)?

First plot: see from one side:

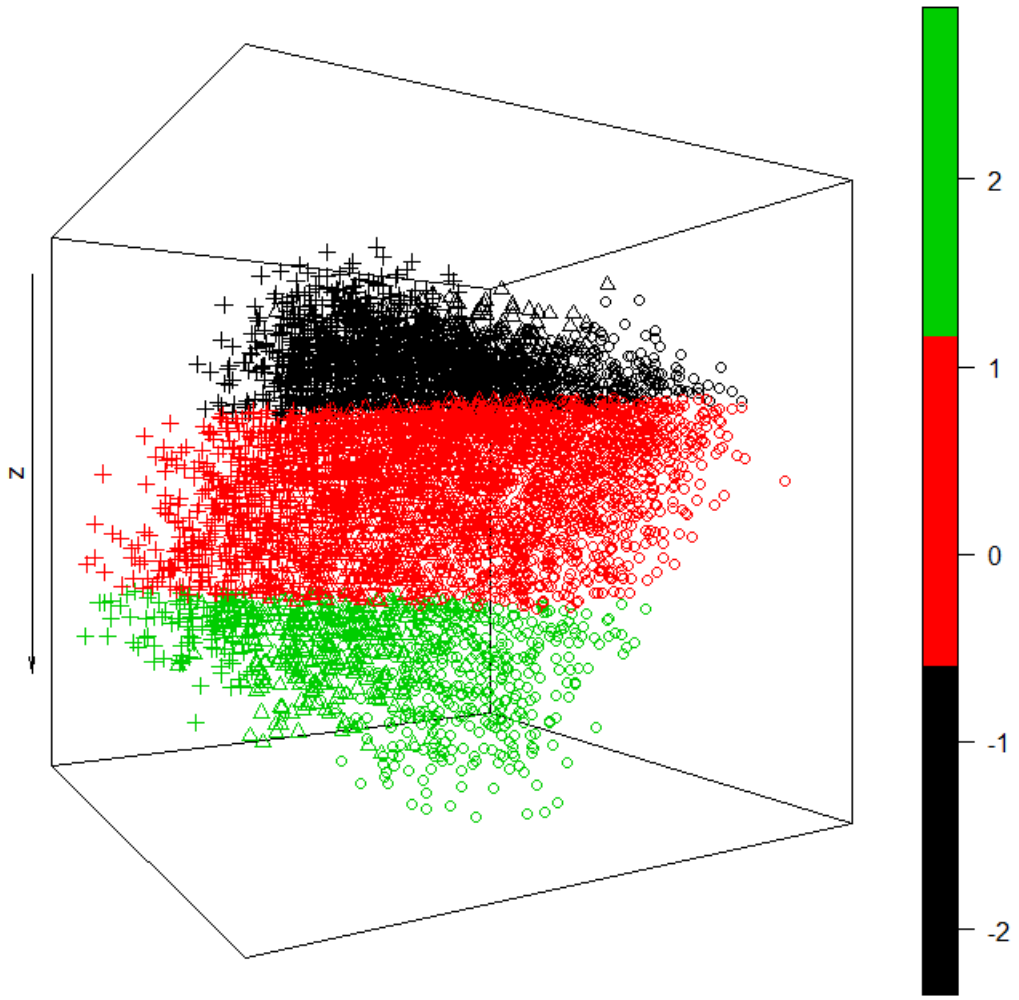


Form some aspects:


z
y
2
1
0
-1
-2



Rotate again:



So, we can find that they are roughly well-separated.

(3)

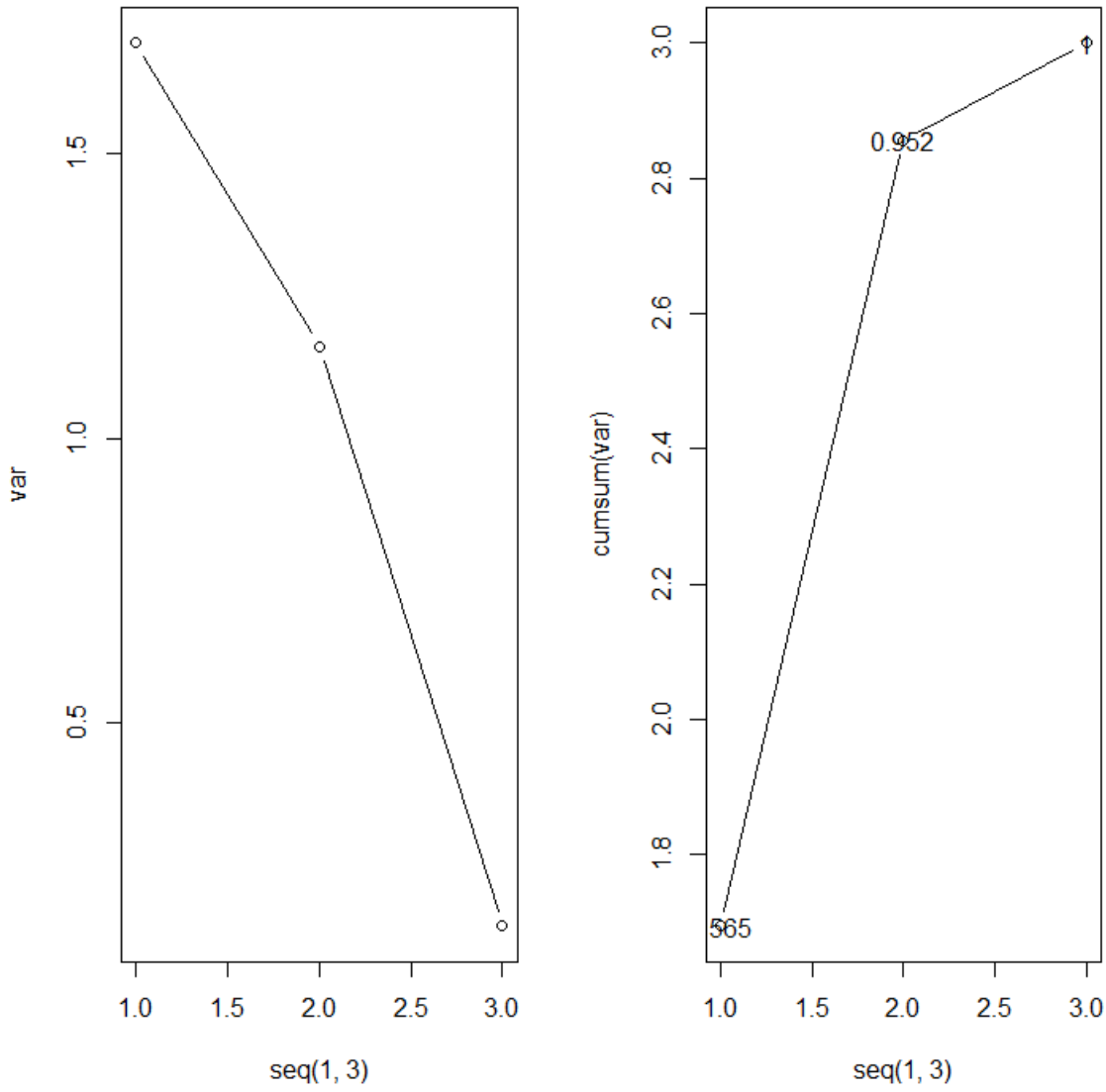
Next, let us do PCA on this dataset.

Result:

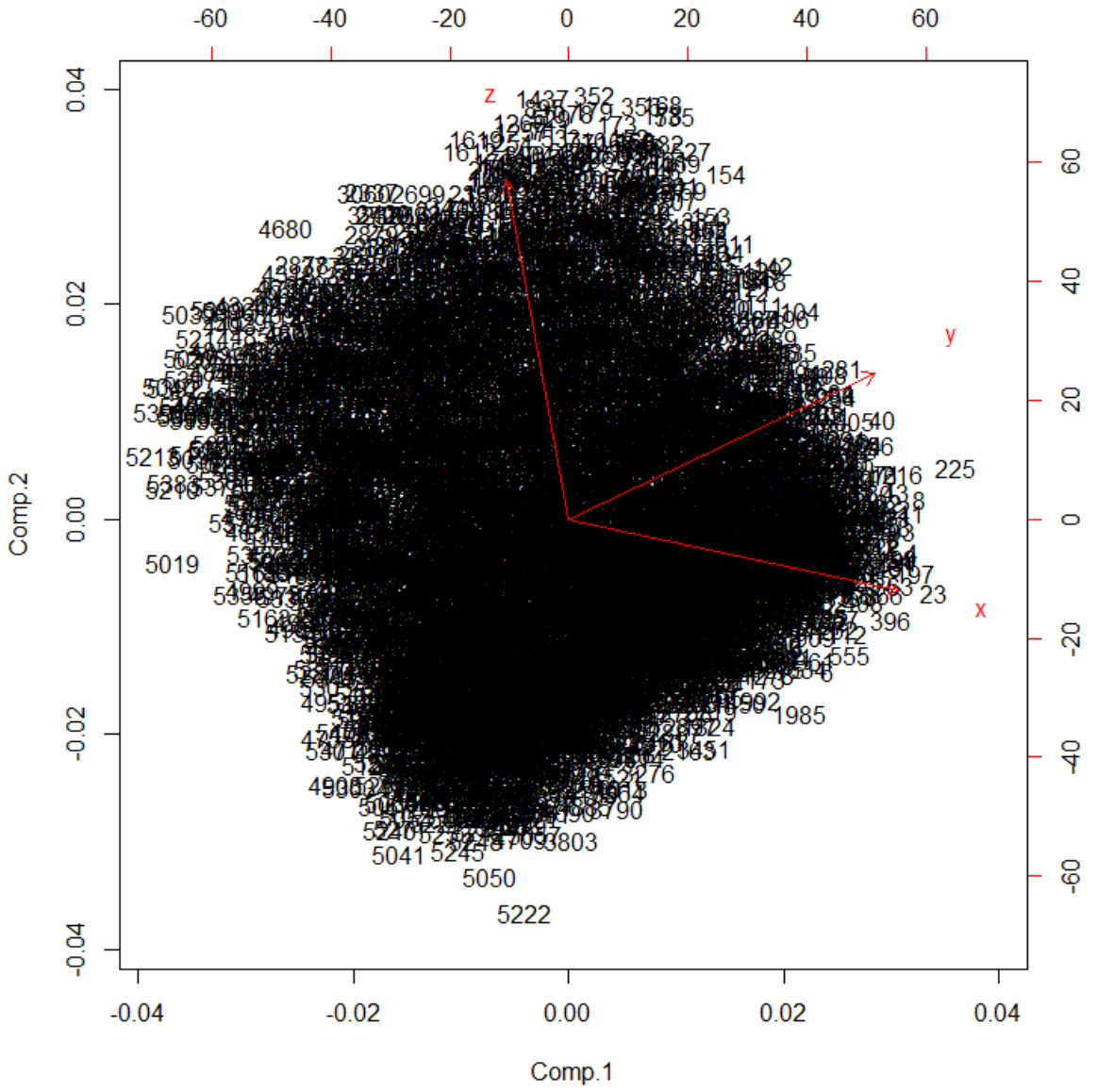
```
> summary(pr.out,loadings=T)
Importance of components:
                          Comp.1    Comp.2     Comp.3
Standard deviation     1.3016432 1.0779248 0.37921378
Proportion of Variance 0.5647584 0.3873073 0.04793436
Cumulative Proportion  0.5647584 0.9520656 1.00000000
```

```
Loadings:
  Comp.1 Comp.2 Comp.3
x  0.727 -0.191  0.660
y  0.673  0.389 -0.629
z -0.136  0.901  0.411
```

PCA scree plots:



Two components seem enough (It follows our intuition: these three groups are well-separated in the space, so a pair index is enough to locate every points in some sense).

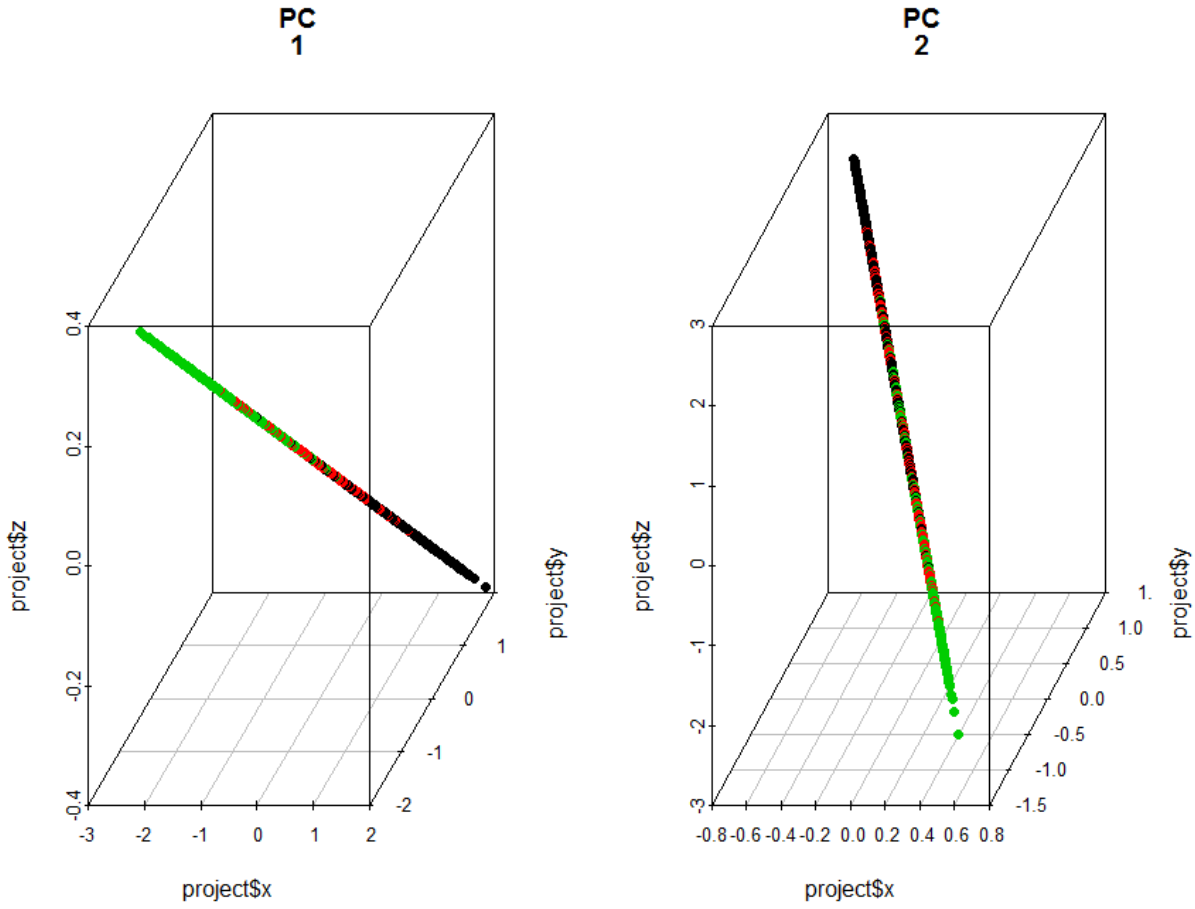


The first component put much weight on x and the second put much on x, and they put roughly equal weights on y.

(4)

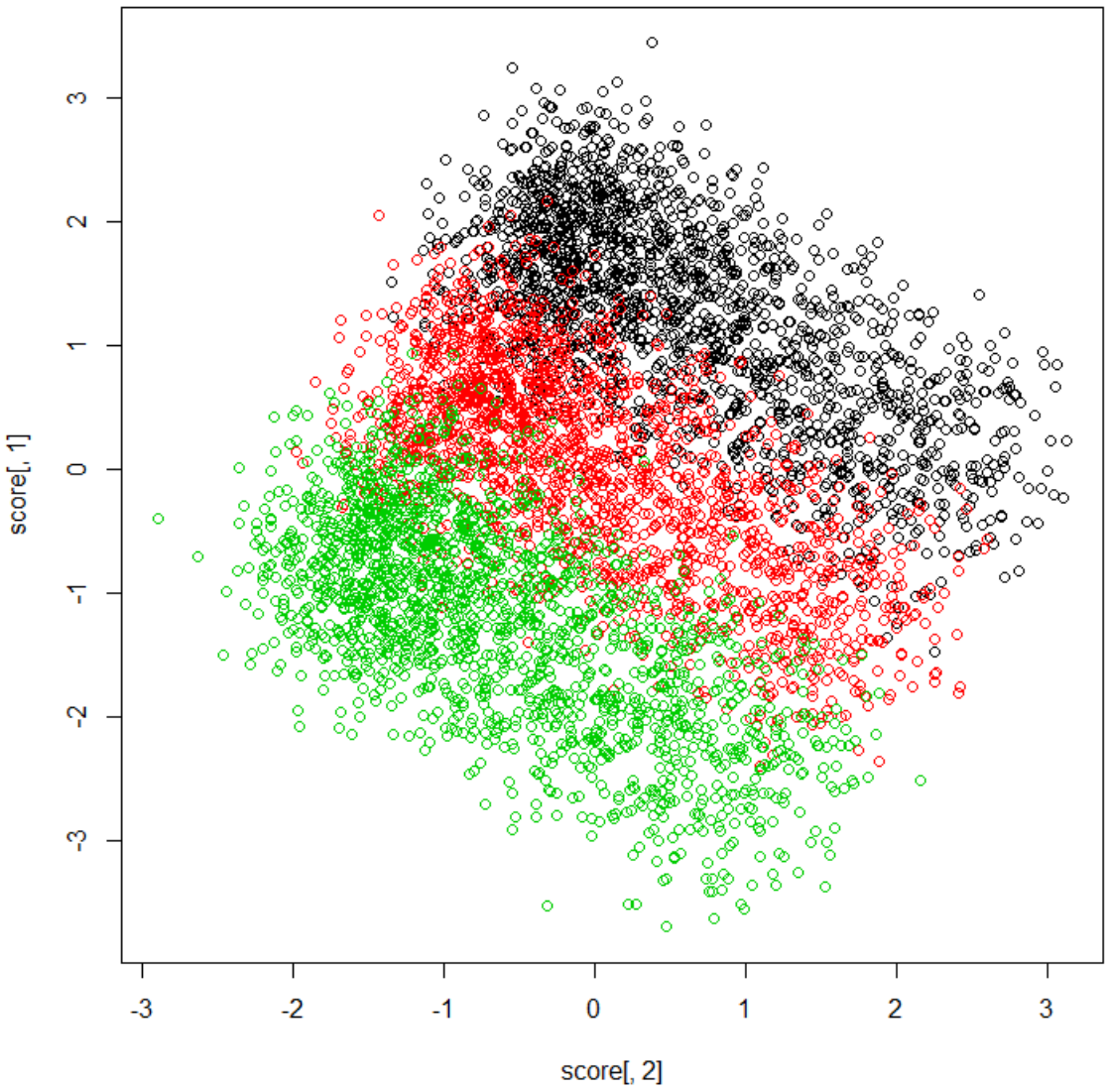
First we take a look at the projection on the first and second principle components:

(denote u1 as first pc, the projection on it is <x-mu,u1'>*u1', and move the projection's center to the original, we get the points mu+<x-mu,u1'>*u1'. So is the second pc).



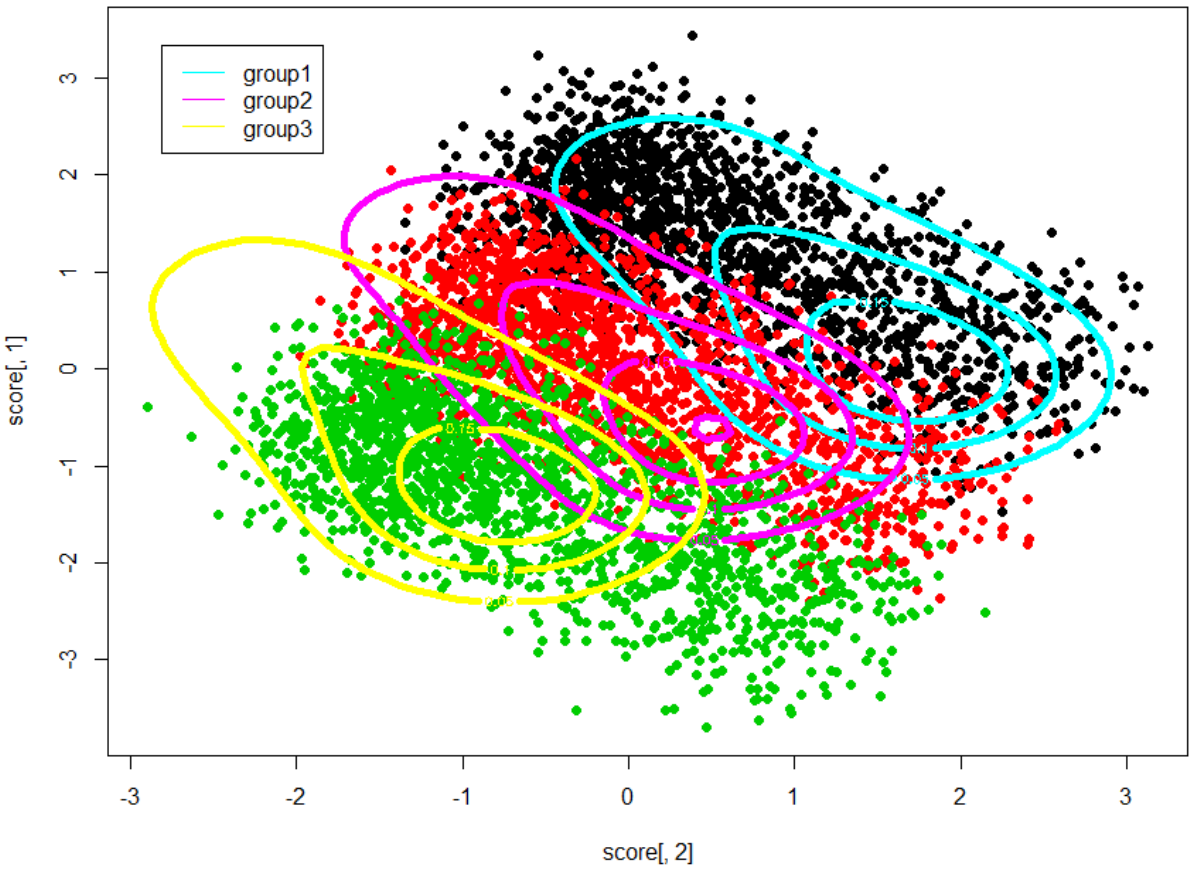
Then we use the score on first two components to recover the data.

Initial recovery in the space stretched by the first two principle components:



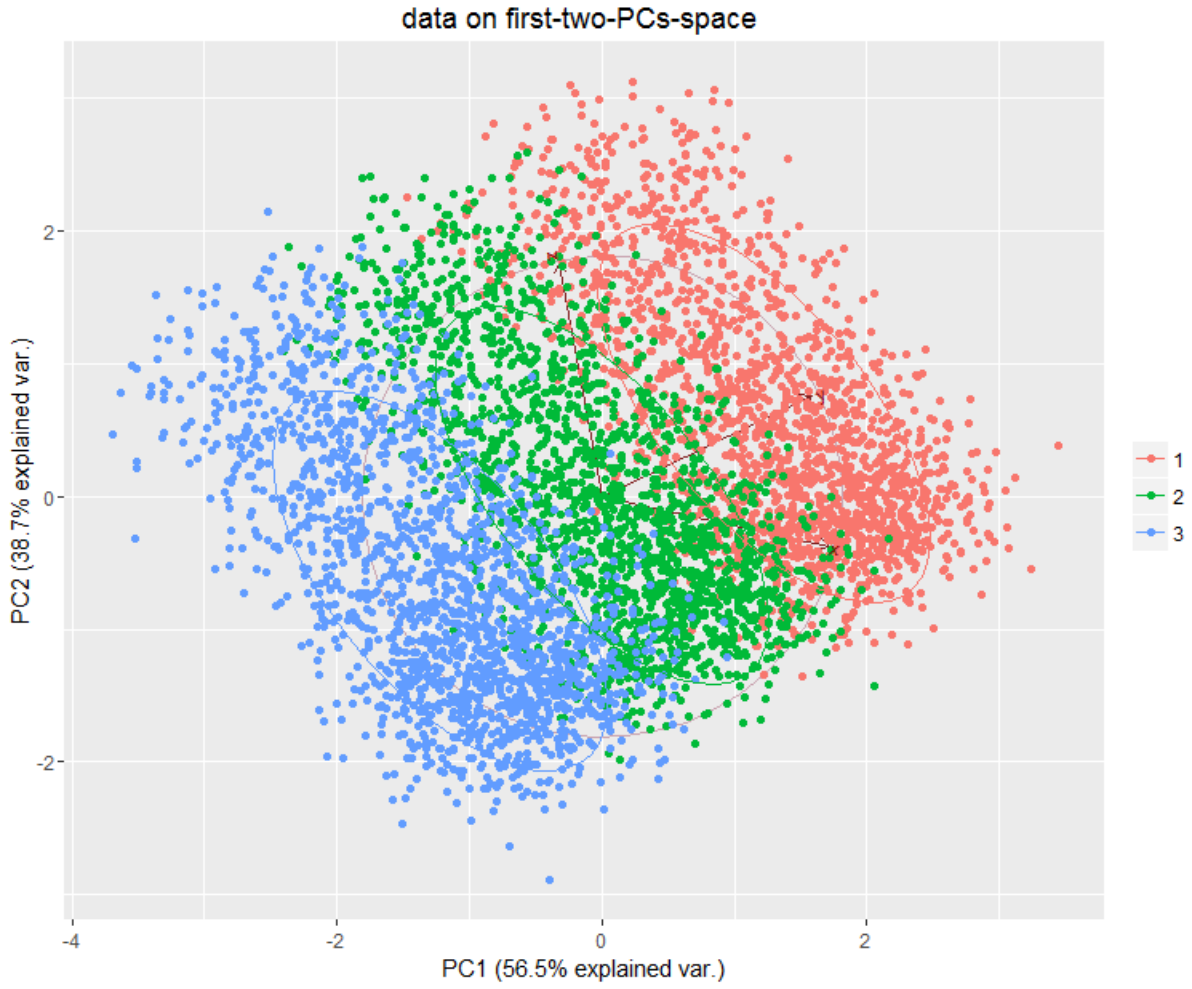
We can find the first two PCs have a good conclusion about the structure of the data (as we analyzed above: 1, not in a plane: the respective structure is like a cloud; 2, well-separated).

The contour plot:



On the space stretched by first two PCs, they (group one to three) have the similar distributions, obviously shown in the contour plot.

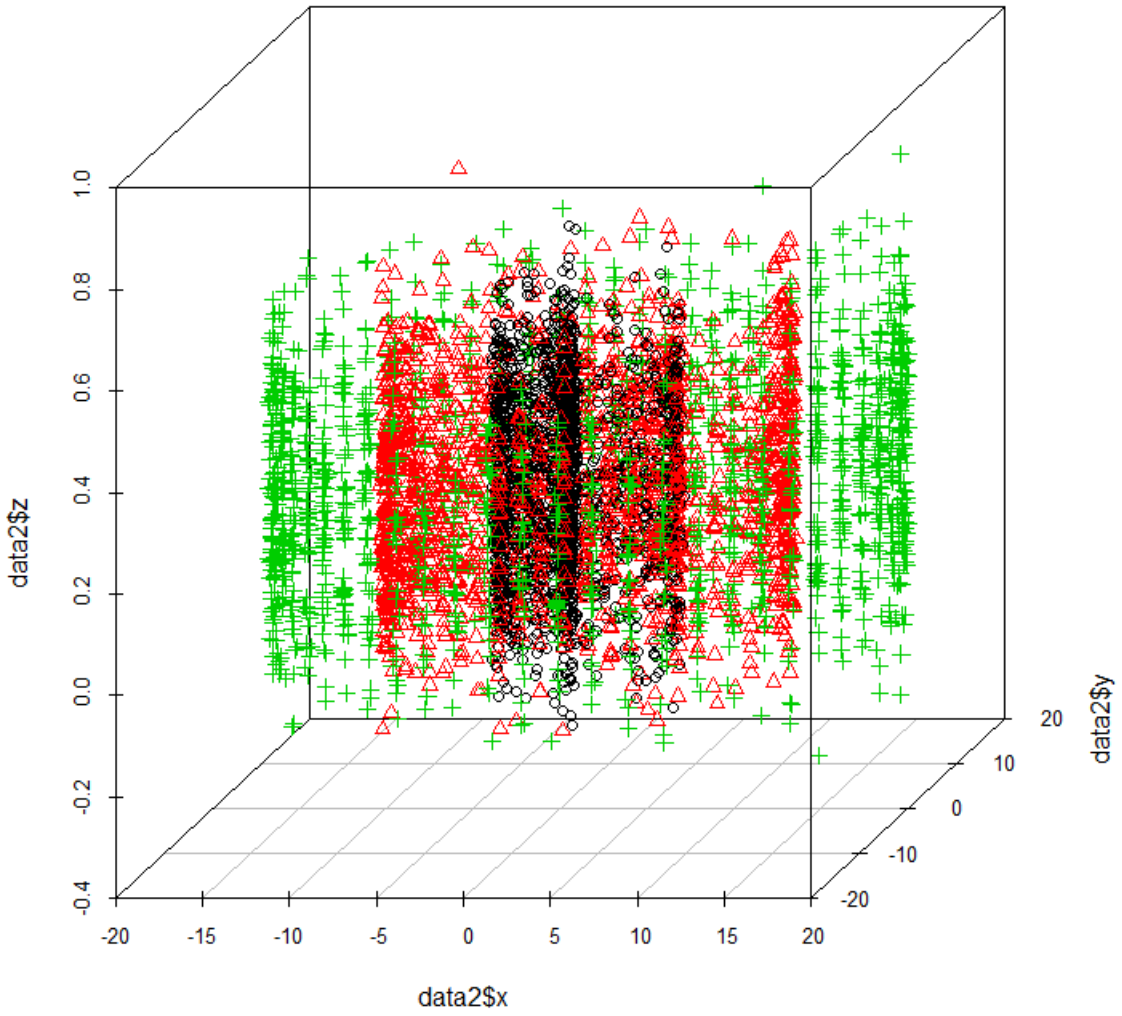
By using ggplot2:



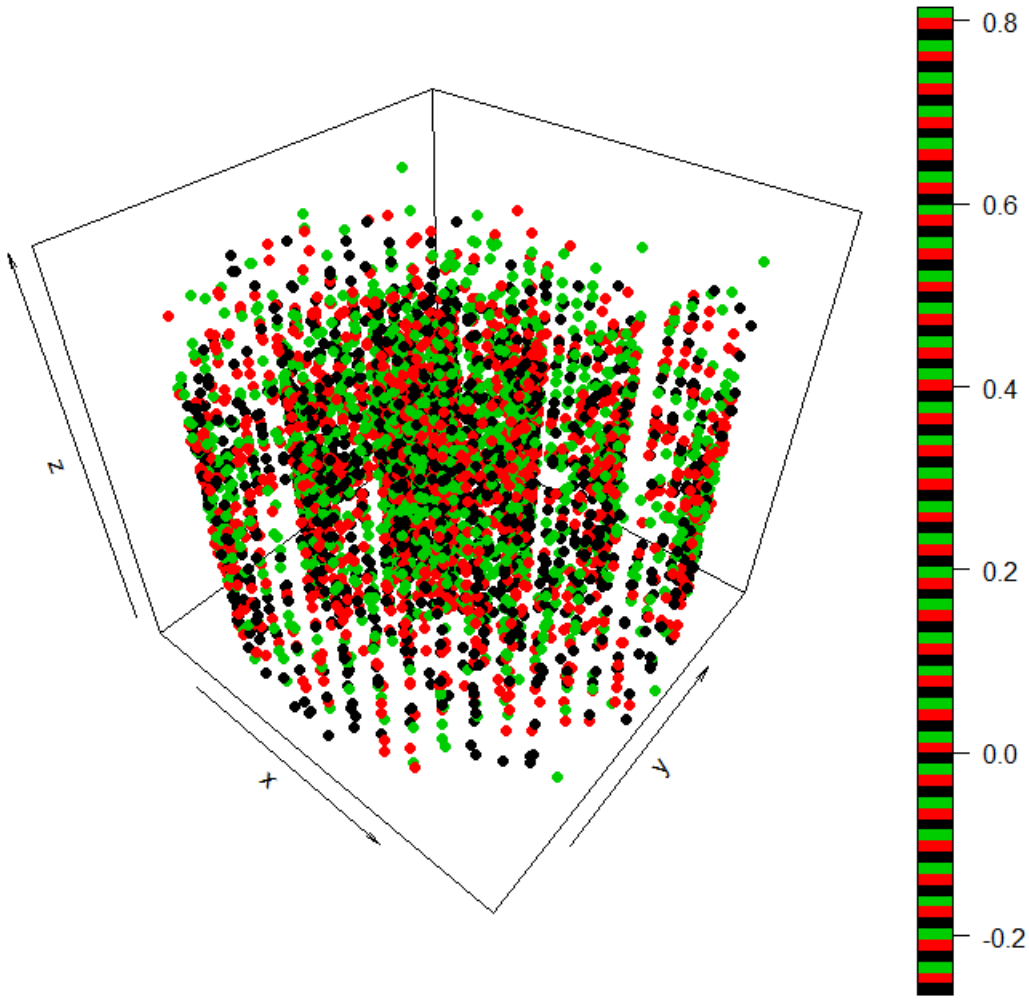
Problem 2:

(1)

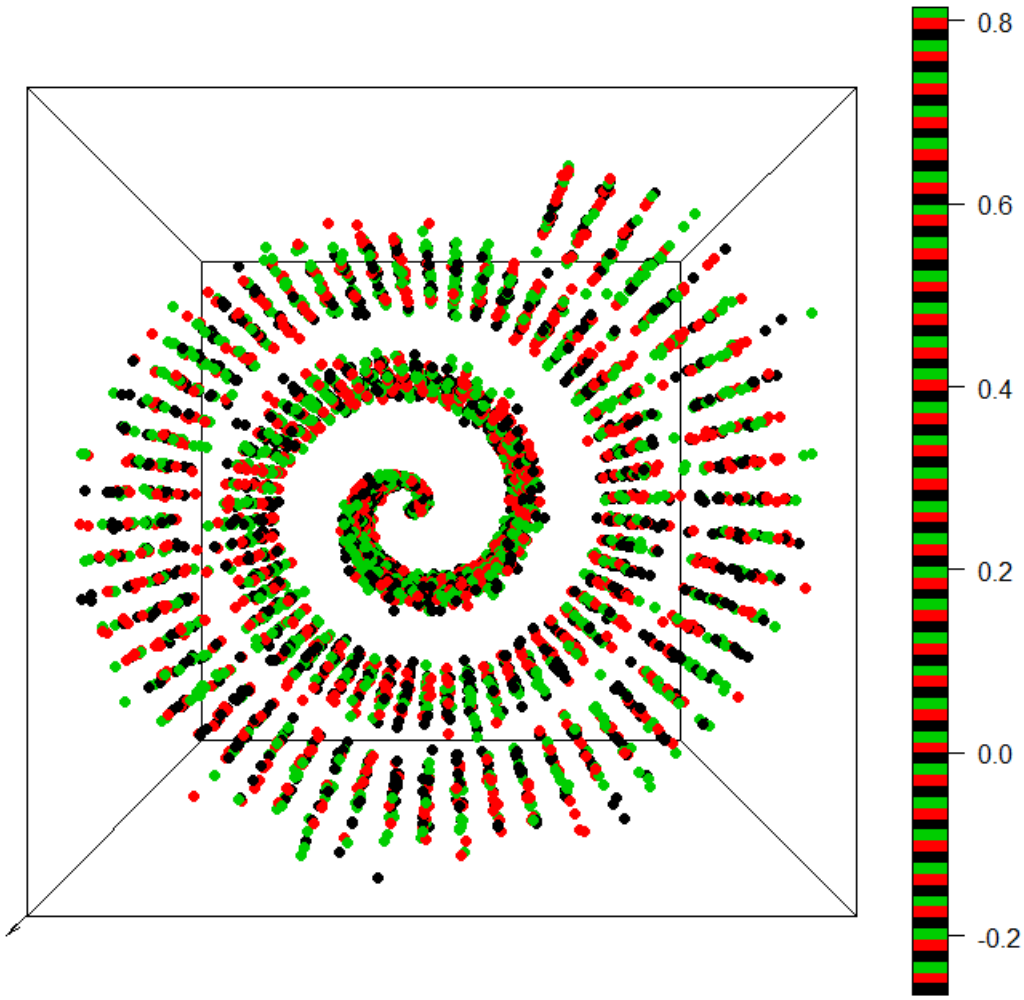
The first 3D plot:

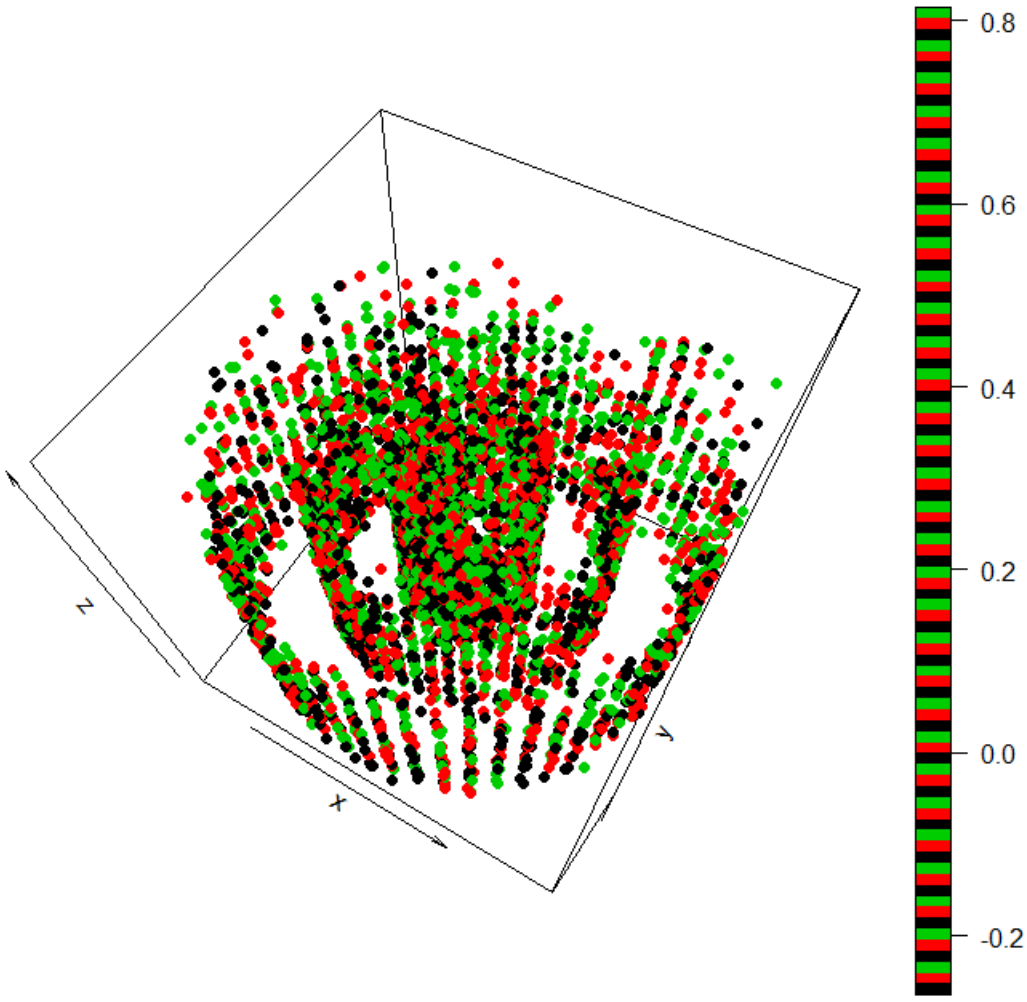

data2$z
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-20
-15
-10
-5
0
5
10
15
20
data2$x
data2$y
20
10
0
-10
-20

Second:

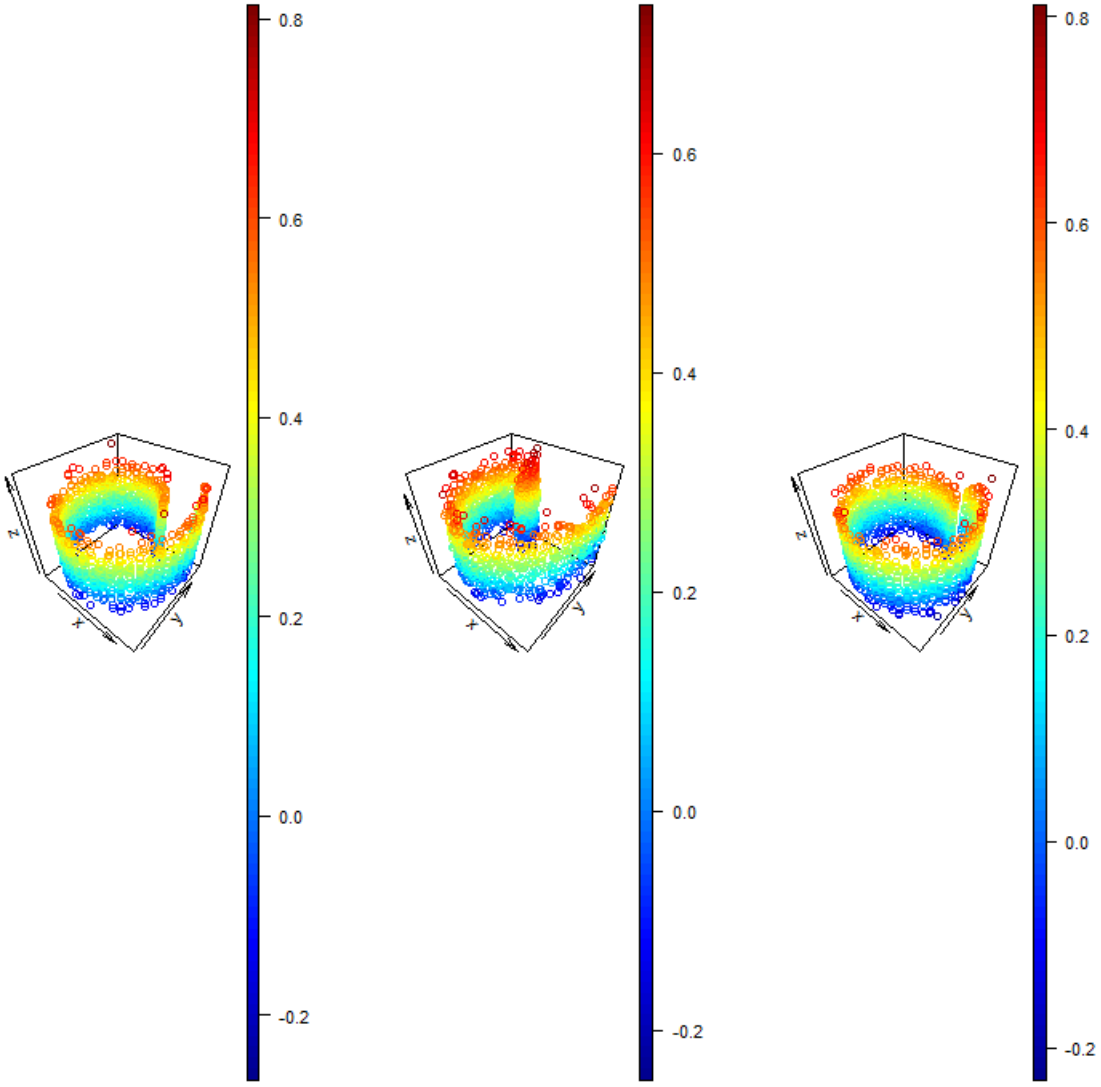


Different angles:


0.8
0.6
0.4
0.2
0.0
-0.2



These three groups:



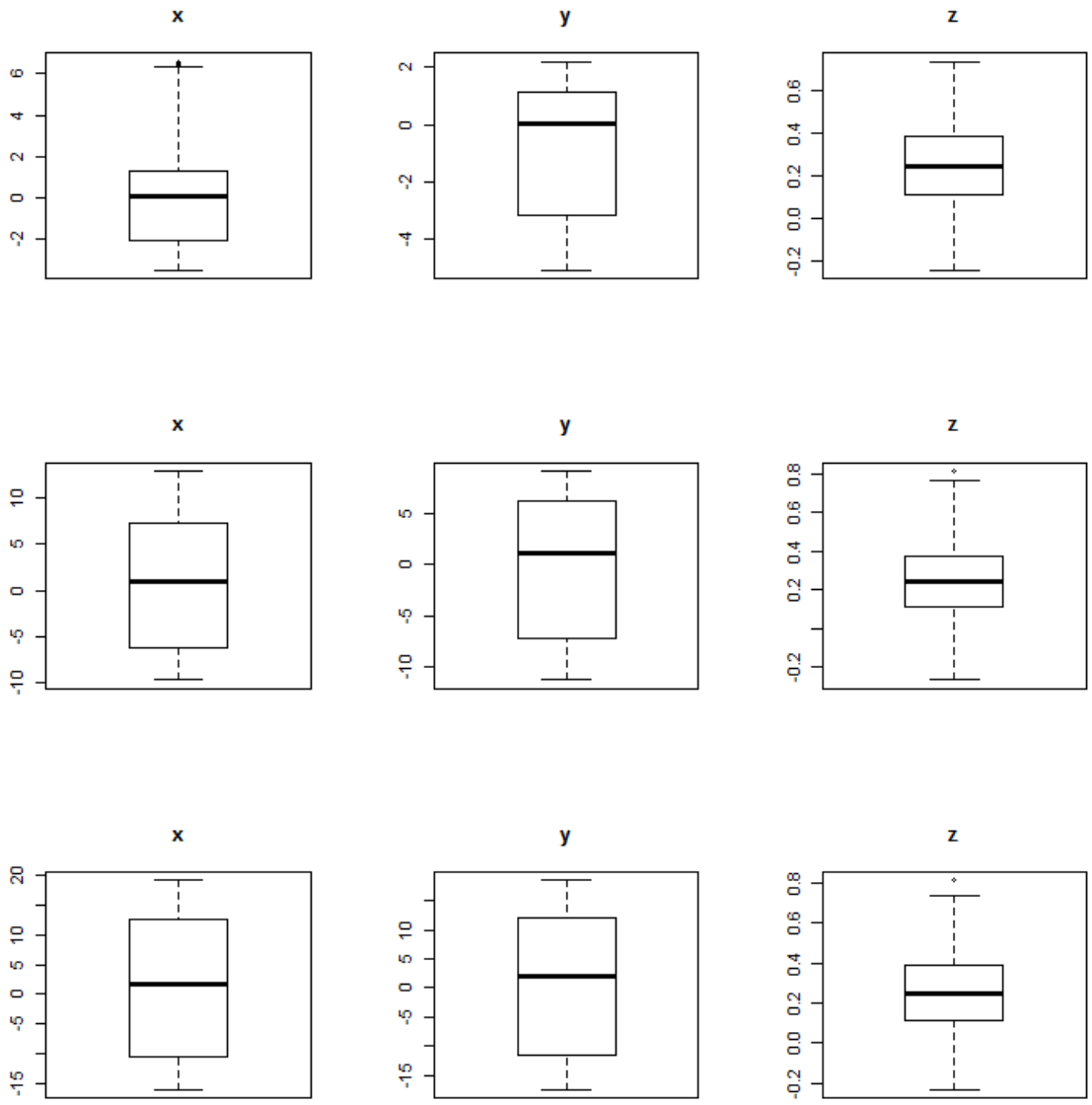
We can find the similar pattern in each structure of different groups: their structure in the space is like a roll!

So, from above plots, we can conclude that:

1) These three groups have similar structures.

2) They are highly mixed.

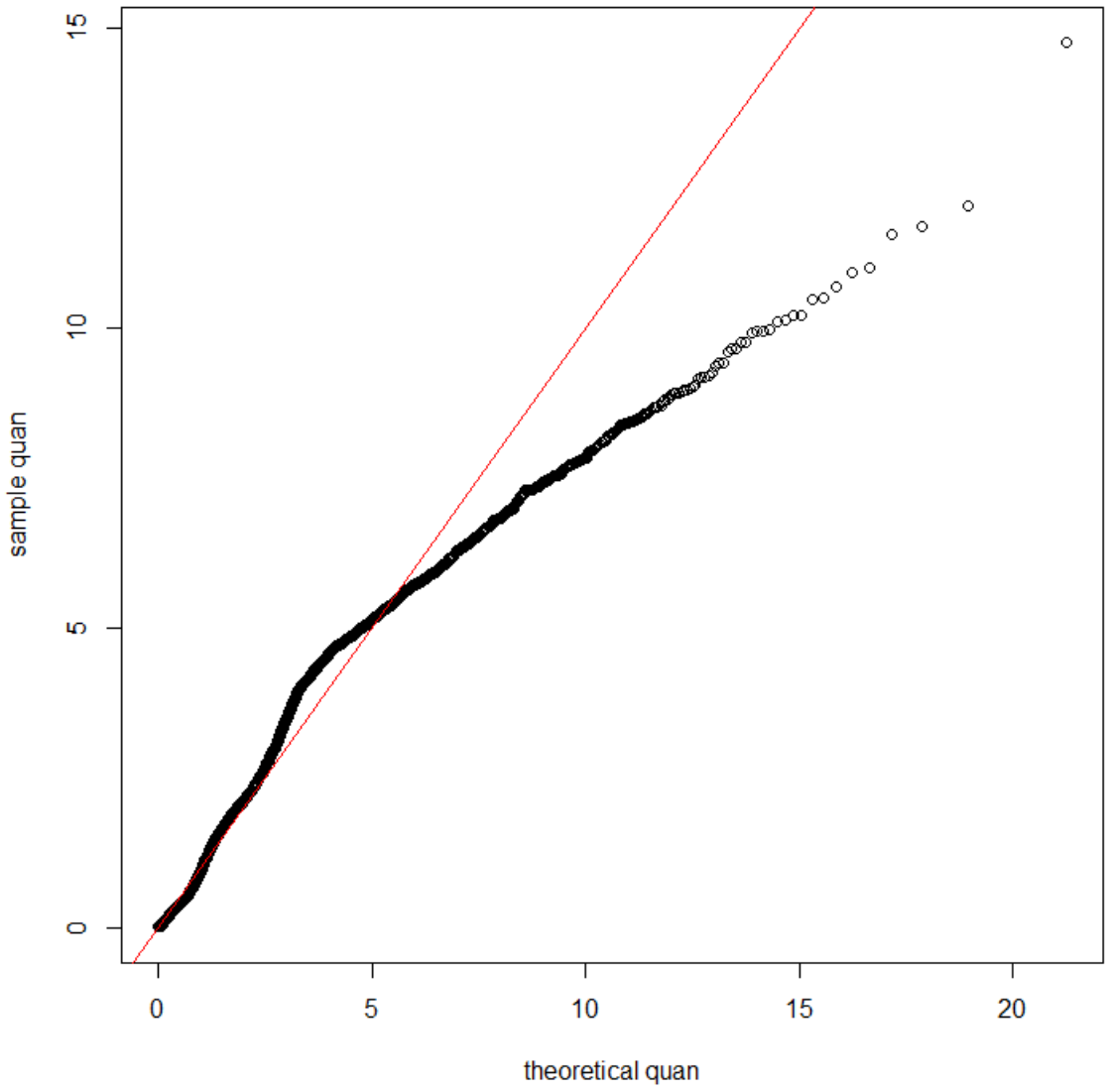
(2)



The x's distribution of group 1 is somewhat wired. And z in group 2 and group 3 might have some outliers. All other variables in each group is plain enough.

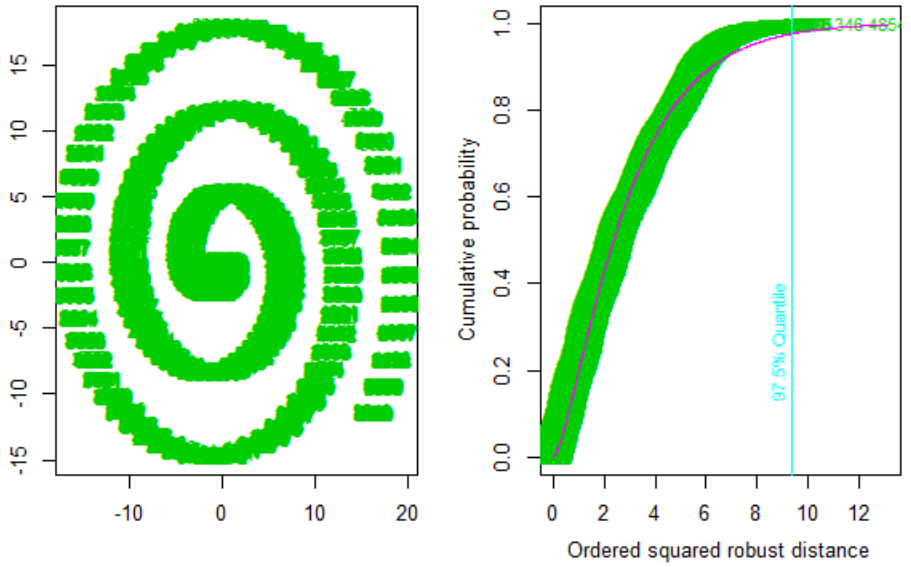
Let us check if these data come from multivariate normal distribution:

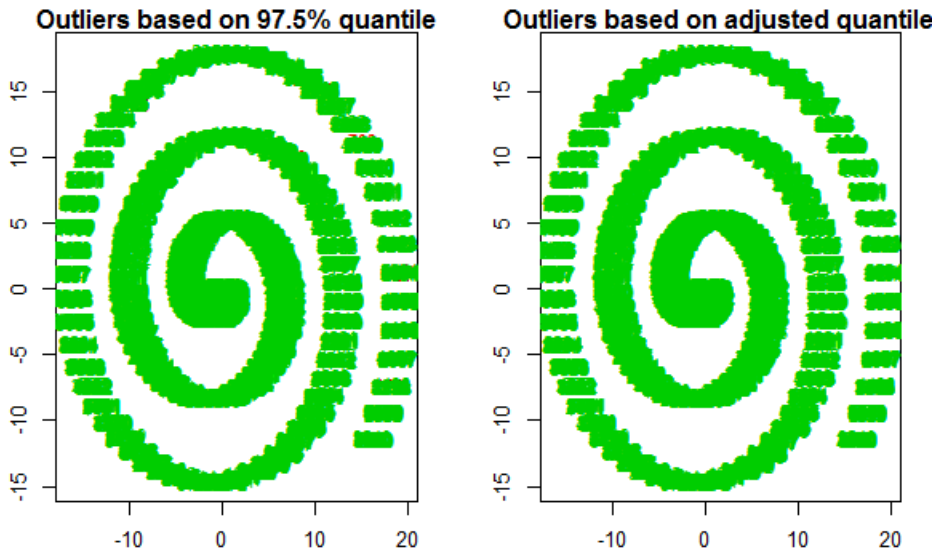
If it comes from the normal distribution, its Mahalanobis distance should distributed as Chi-square with degree of freedom 3. The qqplot is shown below:



We can find there are less points lie in the line, so these three variables are not comes from a multivariate distribution. And we can find from this plot that there might exist some outliers (the heavy tails).

Let us check out the outliers in this dataset:





The result:

```
   > check=aq.plot(data2[,1:3],delta=qchisq(0.975,df=3),quan=0.5,
alpha=0.05)$outliers
    Projection to the first and second robust principal component
s.
    Proportion of total variation (explained variance): 0.9885905
   > for (i in 1:length(check)){
   +   if(check) {pirnt(i)}
   + }
   There were 50 or more warnings (use warnings() to see the firs
t 50)
   > unique(check)
    [1] FALSE
```

There seem no outlier suggested.

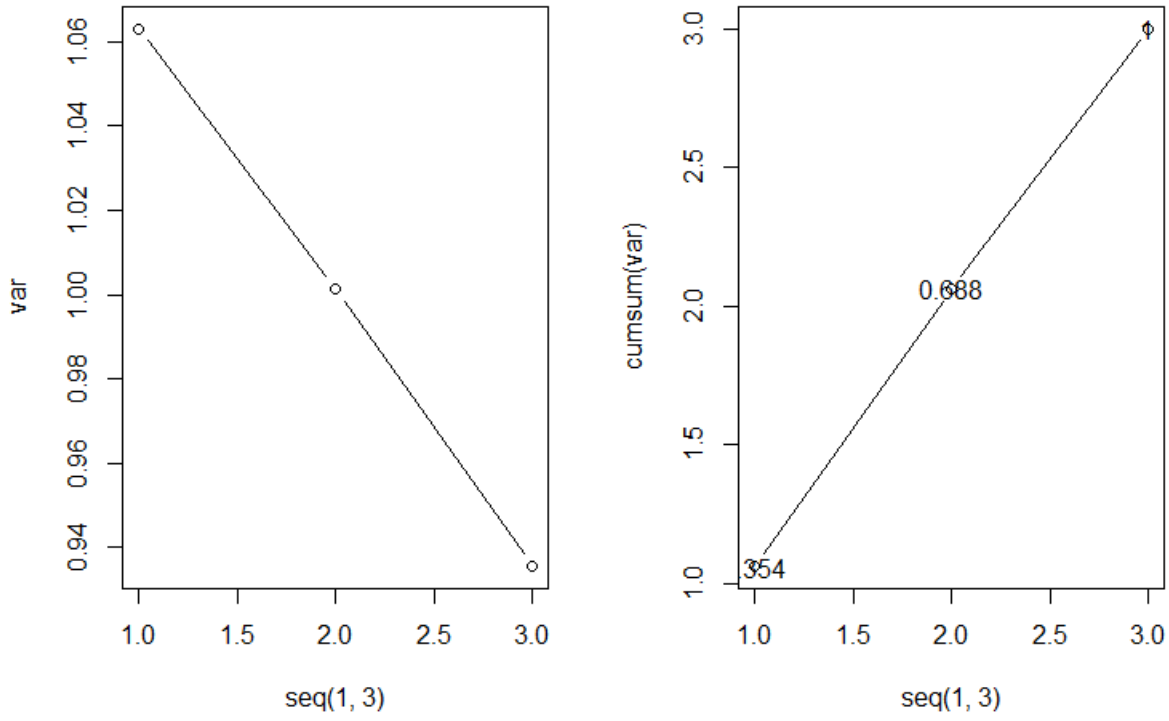
(3)

PCA:

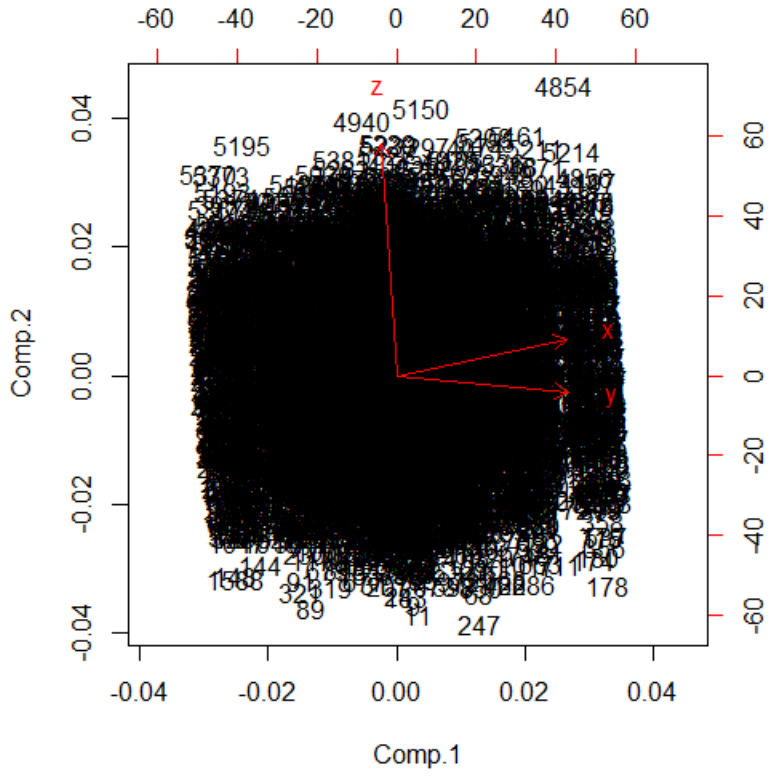
Result:

```
> summary(pr.out,loadings=T)
Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     1.0310453 1.0007093 0.9672262
Proportion of Variance 0.3543515 0.3338063 0.3118422
Cumulative Proportion  0.3543515 0.6881578 1.0000000

Loadings:
     Comp.1 Comp.2 Comp.3
Var1  0.702  0.155  0.695
Var2  0.709        -0.702
Var3         0.986 -0.158
```

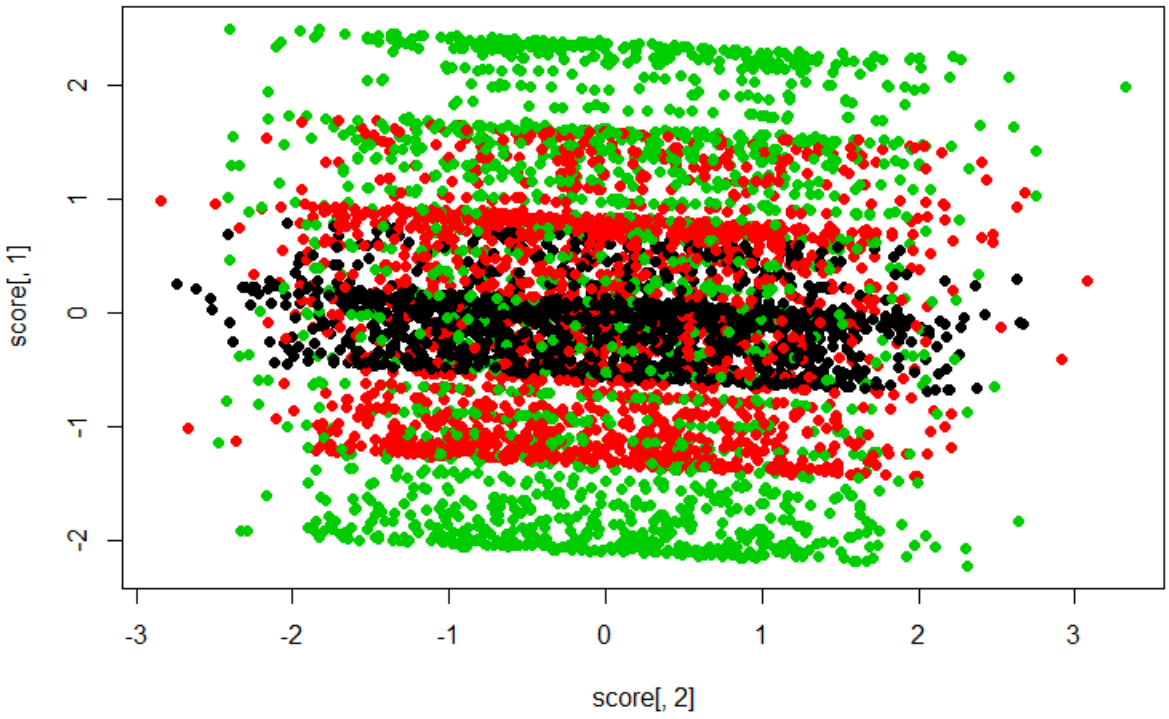
Scree plots :



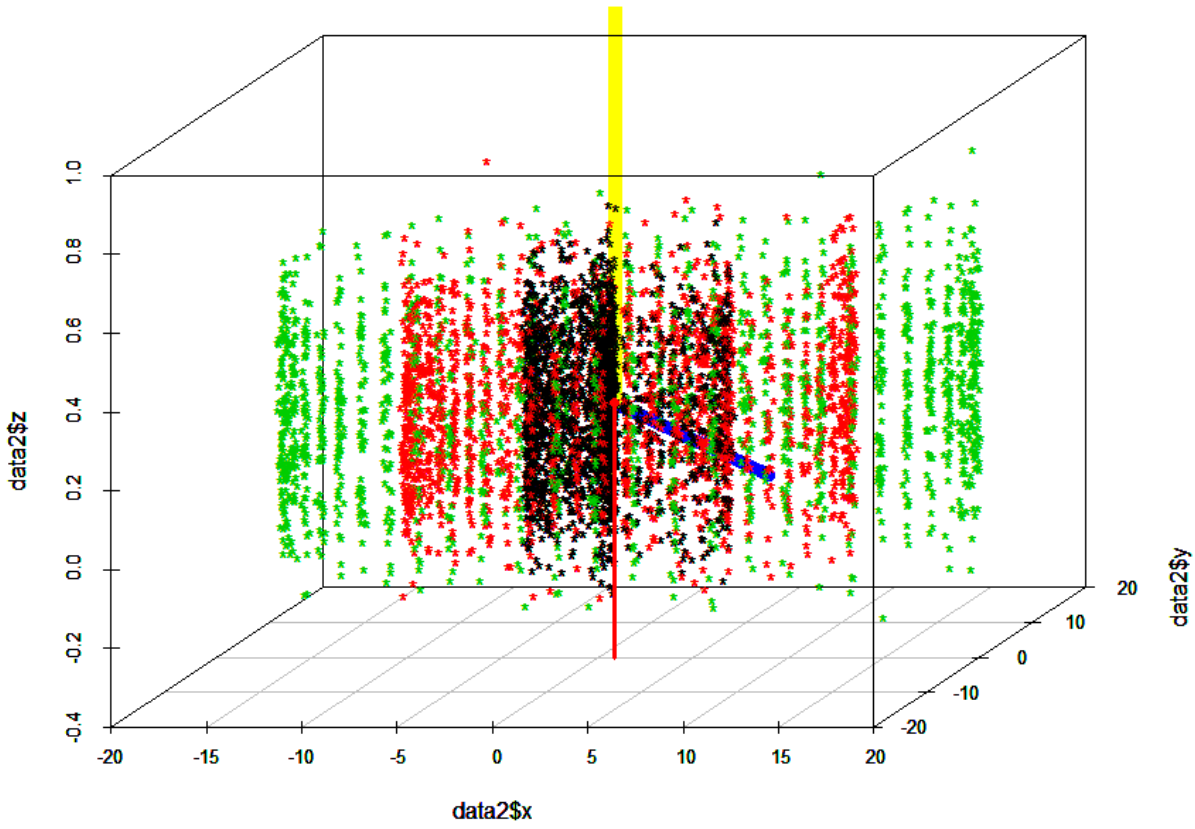
it seems first two components will not have a good summary of the original data.



The biplot shows the second PC put much weights on z and the first put much weights on x and y.



We can see from the 2-d plot recovered by first two PCs.



The red point(with a line to the xy-plane) is the mean point of this dataset, and the blue line is the first PC's direction, the yellow one is the second PC's direction.

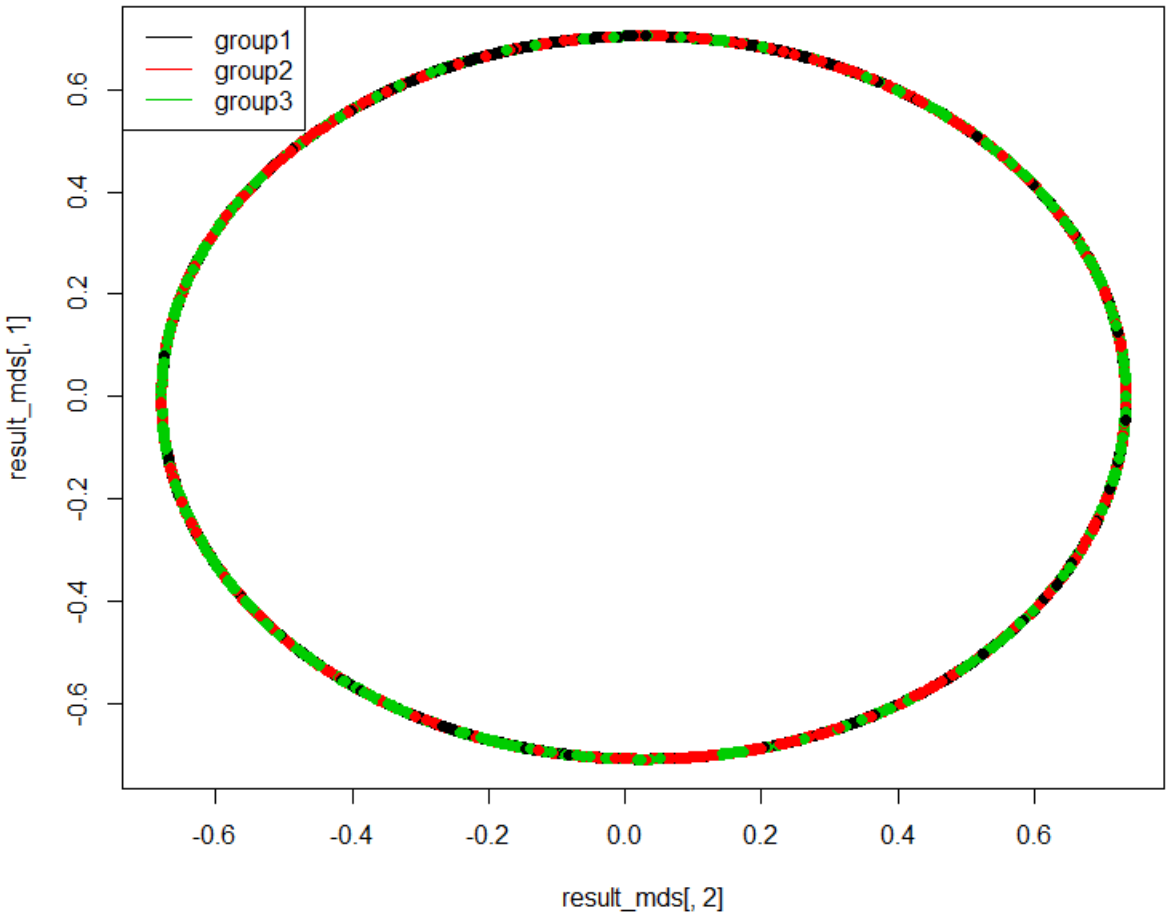
(4)

Let us first do MDS on the dissimilarity matrix converted by the correlation matrix by using the centered and scaled data.

And we recover the data space by using the two-dimension data. The recovered data space is shown below:

```
> head(mds$eig)
[1] 1.420995e+03 1.275040e+03 4.850677e-12 4.139093e-12 4.018676
e-12 3.434046e-12
```

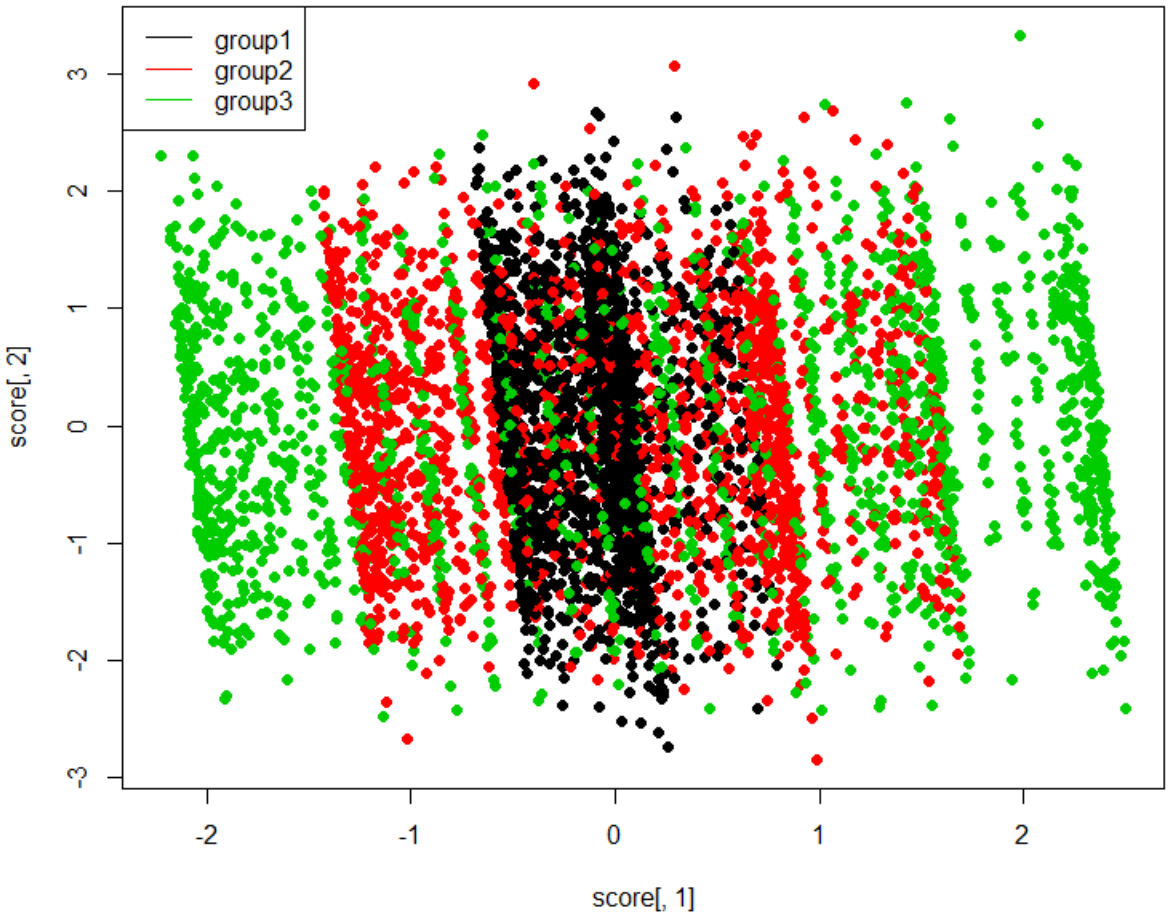
The first two eigenvector is enough to act as index for the data recovery.



And then we do PCA on this data set. And recover the data space in the space sketched by first two PCs.

And the result is shown below:

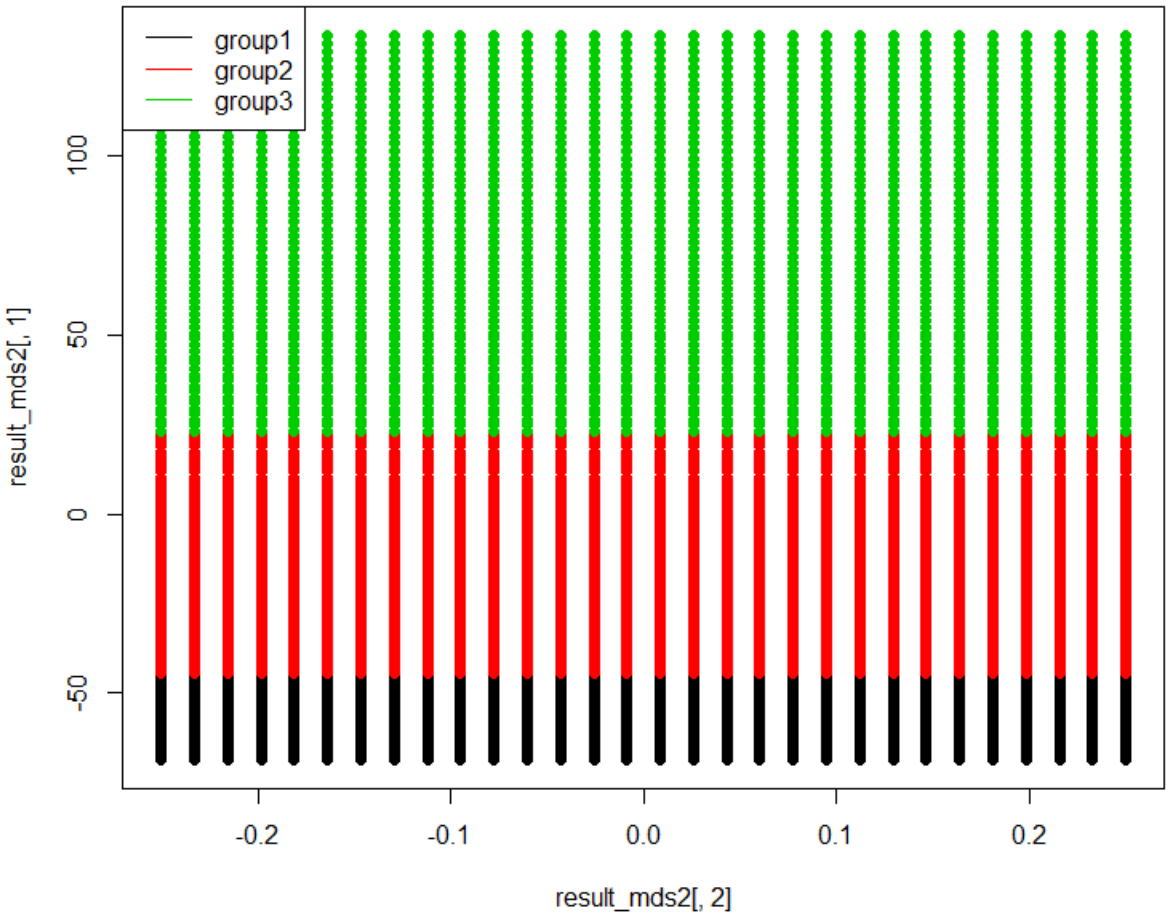
```
summary(princomp(data2[,1:3],cor=T))
Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     1.0310453 1.0007093 0.9672262
Proportion of Variance 0.3543515 0.3338063 0.3118422
Cumulative Proportion  0.3543515 0.6881578 1.0000000
> pr$loadings

Loadings:
  Comp.1 Comp.2 Comp.3
x  0.702  0.155  0.695
y  0.709        -0.702
z         0.986 -0.158
```



By using the dissimilarity matrix, we get:

```
> head(mds2$eig)
[1] 1.967648e+07 1.202586e+02 8.704511e-08 6.817707e-08 4.139510
e-08 4.080598e-08
```

It seems the first two eigenvector is enough to have a good recover of the data.



(5)

Conclusion:

As we can see, if we use correlation matrix as similarity matrix, it can recover some characters of the data, like highly-mixed, like-roll structure, but it is a poor recovery because the character of relative position of each group is missing. We can have a think about this result, why correlation similarity matrix result in a plot like this? If we don't take the group distinction of these points into consideration, the correlation between each points are given by (1/3)*<xi,xj>, where xi and xj is the coordinates of each data points. Because the whole data structure is a circle (but the data points lie on and within the circle, asymmetric in some sense), which intuitively give us a circle-structure recovery.

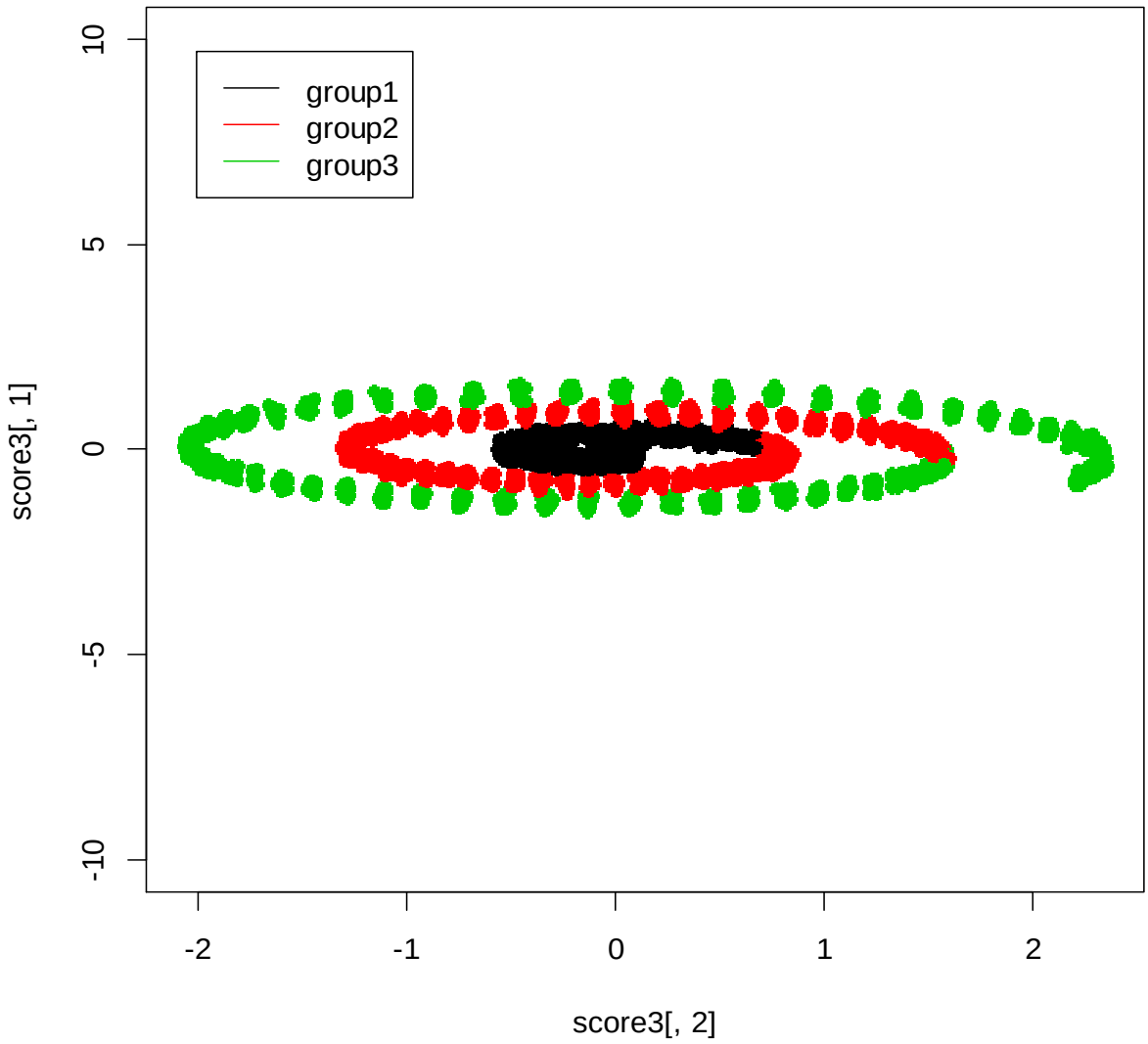
As for the PCs, the second PC put much weights on z and the first put much weights on x and y (partial of x and y not all, we can see from the bi-plot, the first PC's direction is between x and y), so the second PC capture most information about the relative position of each group and the first PC capture most information about the relative position of the points within each group. So the result is it has good recovery about the structure of the whole data. (i.e highly-mixed)

The given similarity matrix (without know how it is derived, so we cannot analysis it), give a good comprehension of the unfold structure of the data: plane, and a good recovery of the relative position within each group. But it loses the information of how the original data is mixed.

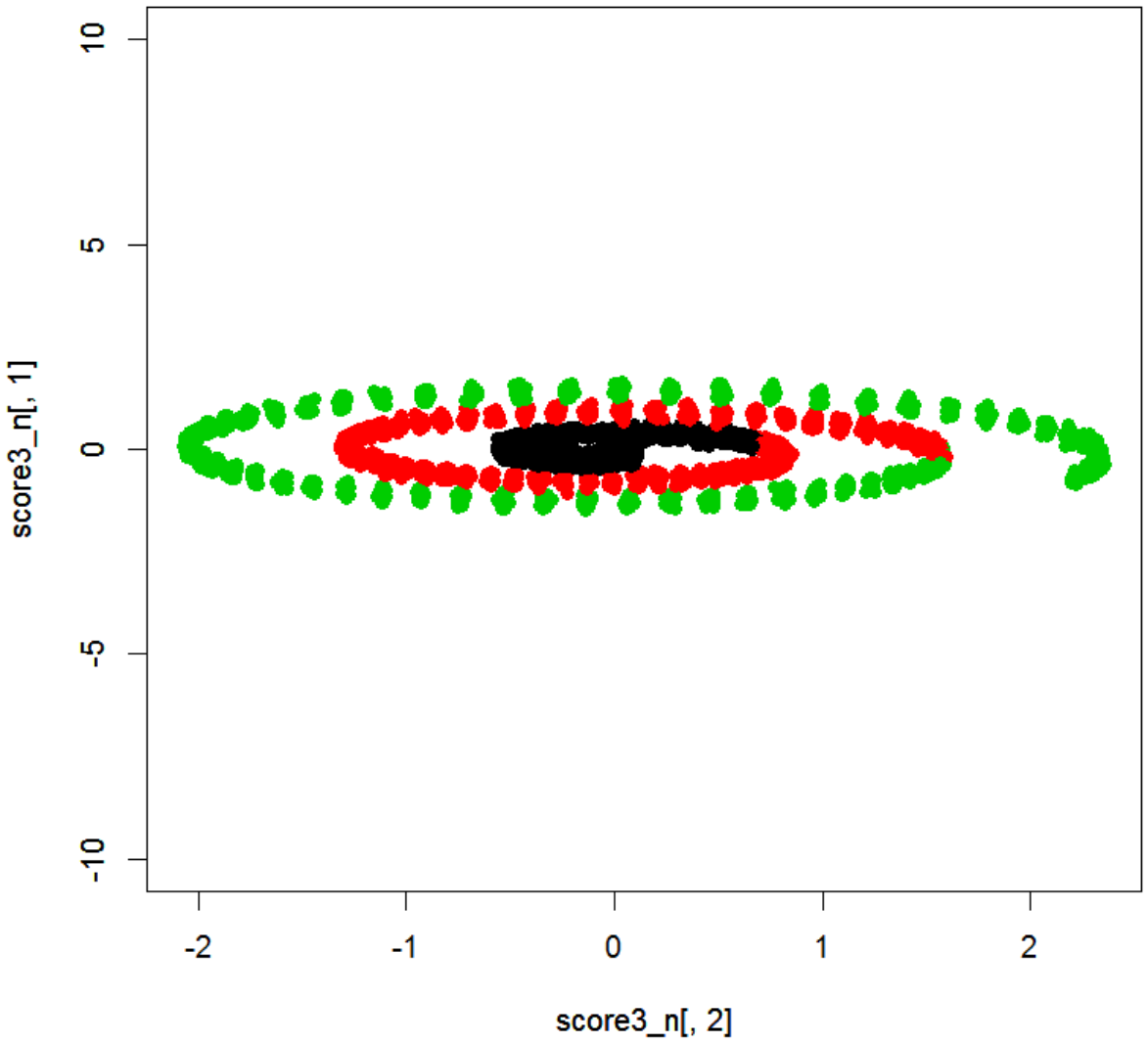
(6)

```
summary(pr.out3,loadings=T)
Importance of components:
                          Comp.1    Comp.2    Comp.3
Standard deviation     1.1022249 1.0166669 0.8668846
Proportion of Variance 0.4049665 0.3445372 0.2504963
Cumulative Proportion  0.4049665 0.7495037 1.0000000

Loadings:
  Comp.1 Comp.2 Comp.3
x  0.471  0.718  0.512
y -0.492  0.696 -0.524
z -0.732         0.681
```

Adding point plot: (include the adding points)

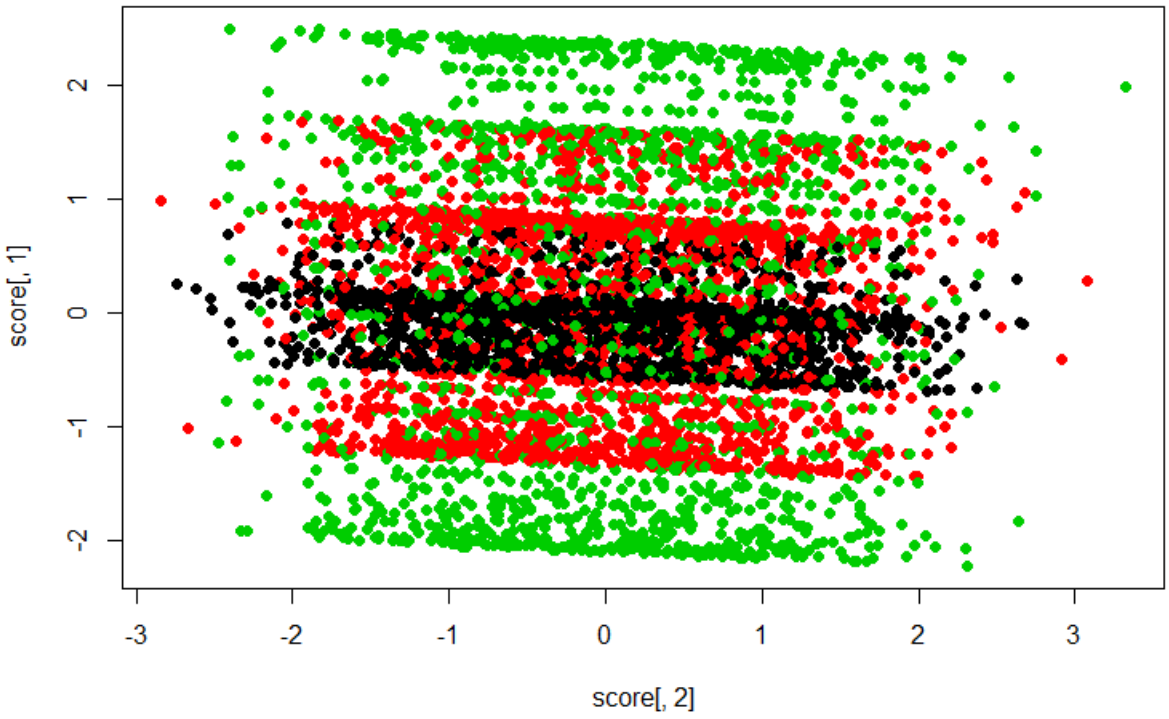


Adding point plot: (exclude the adding points)



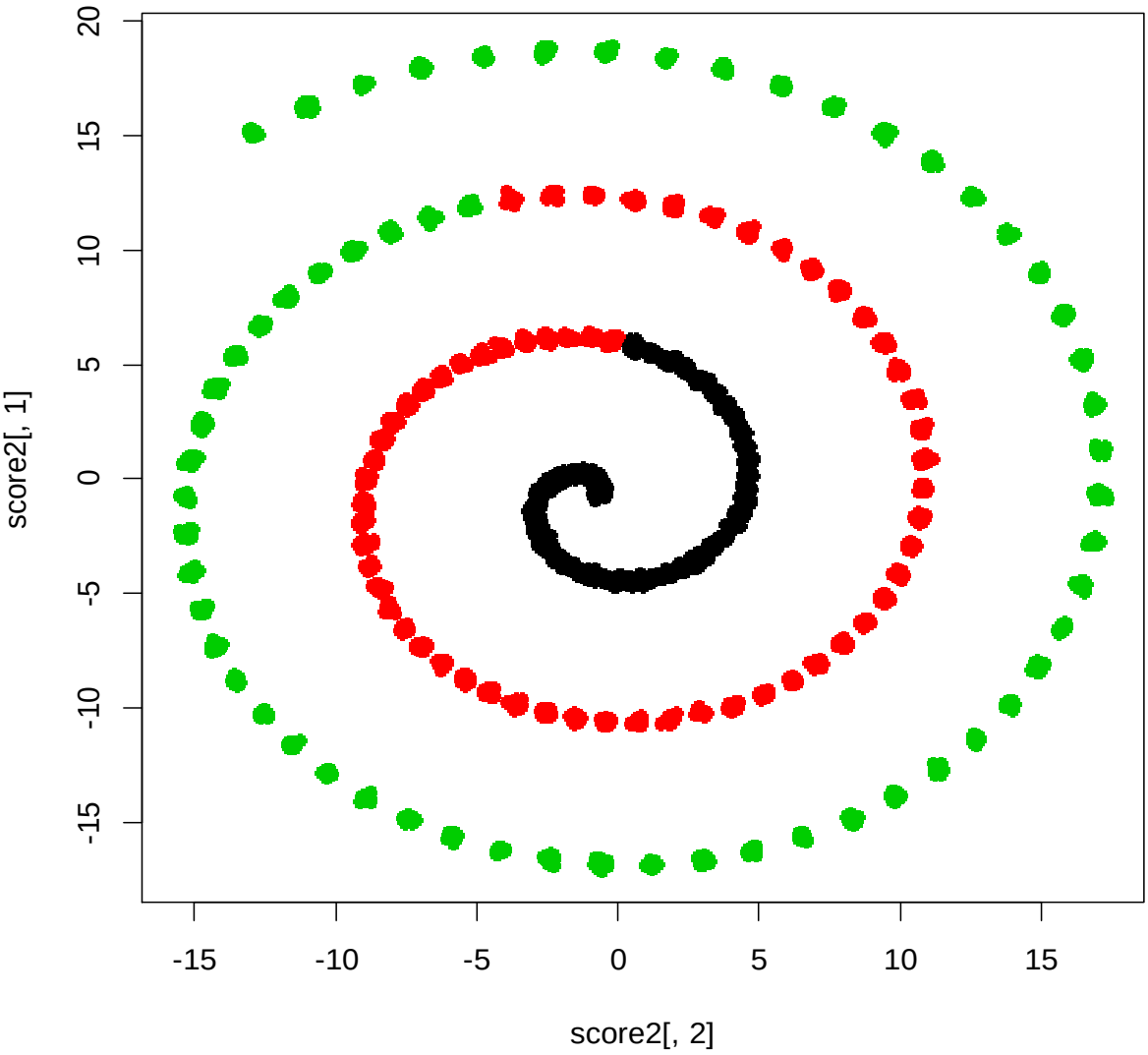
No so much change compared to the plot where there is the adding points.

Recovered by first two PCs by using correlation matrix.



Recovered by first two PCs by using covariance matrix:

```
> summary(princomp(data2[,1:3],cor=F),loadings=T)
Importance of components:
                          Comp.1    Comp.2       Comp.3
Standard deviation     8.451107 7.8359027 0.1791599829
Proportion of Variance 0.537589 0.4621694 0.0002416046
Cumulative Proportion  0.537589 0.9997584 1.0000000000

Loadings:
  Comp.1 Comp.2 Comp.3
x  0.882  0.472
y  0.472 -0.882
z                1.000
```



When compared them, we find:

When adding this strange point (might be viewed as outlier in some sense), the result of PCA have change a lot, as we can see from the adding point result, the first two PCs are very different from the one without this point. This makes the image projected on the space stretched by the first to PC's change a lot! It is somewhat a squeezing image of that generated by PCs got by covariance matrix.

Why they are similar?

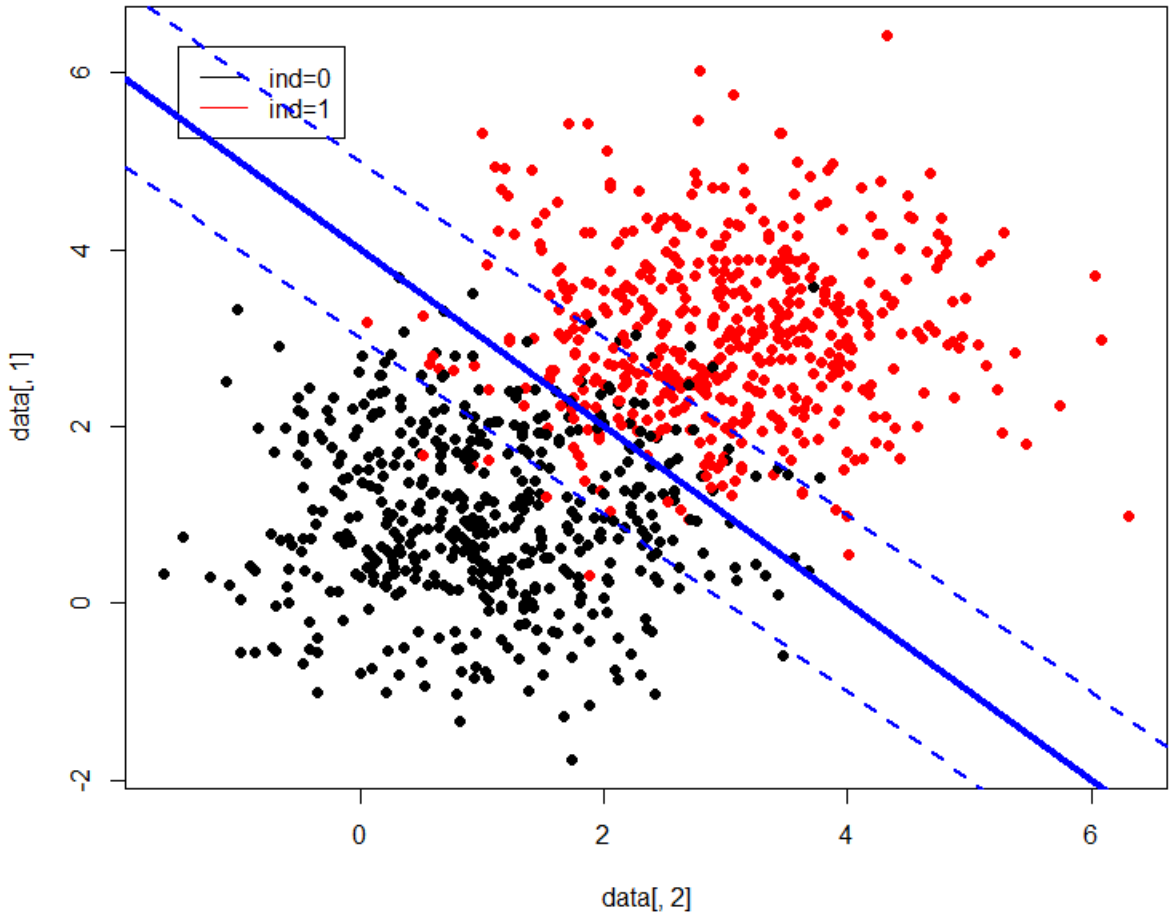
PC captures the most various direction of the data. Group 3 is the most various one and then the second group and then the first group. When adding the strange point, means in one direction, the variance's part of the total variance has increased, and then this direction capture more variance of the data, this is why when adding the strange point, the first PC has a larger part of the total variance. And this is why

the recovery generated by covariance matrix and the correlation matrix when adding the strange points looks the same.

Problem 3:

(1)

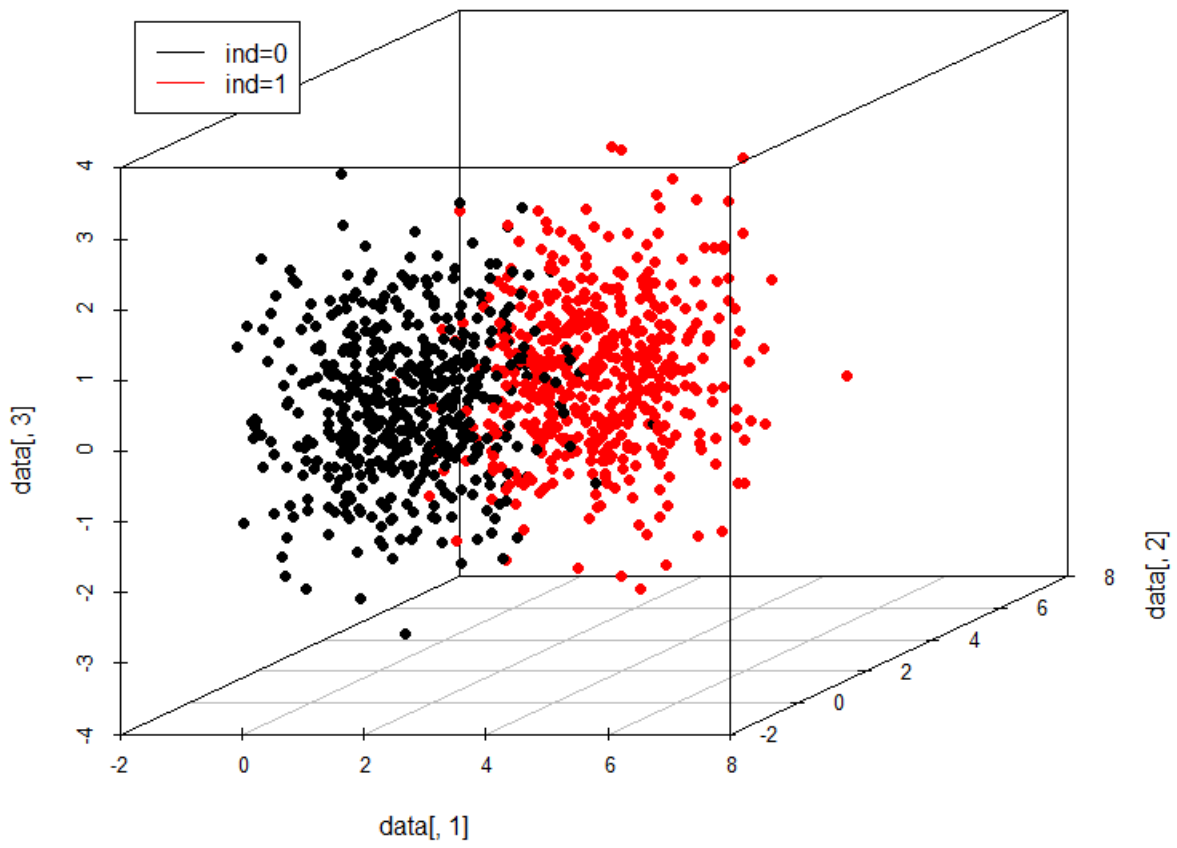
The first and second variables:



We use the code like:

```
> for(i in seq(1,90,length.out=10)){
+   for(j in seq(1,90,length.out=10))
+   {scatter3D(data[,1],data[,2],data[,3],col=data$ind+1,pch=16,
phi=i,theta=j);
+     Sys.sleep(0.1)}
+ }
```

To see the 3-d plot in all angels:

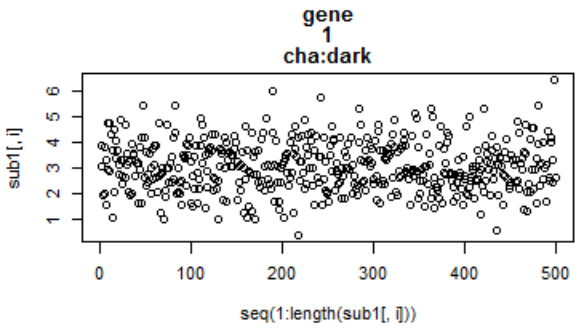


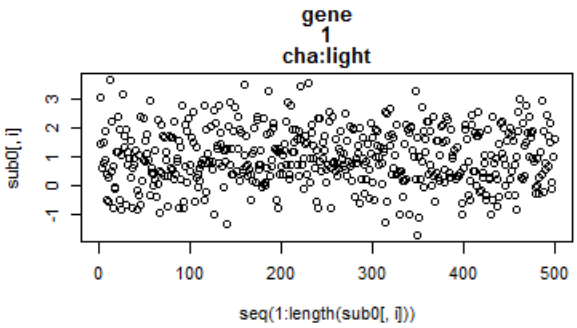
And we find they are not so mixed.

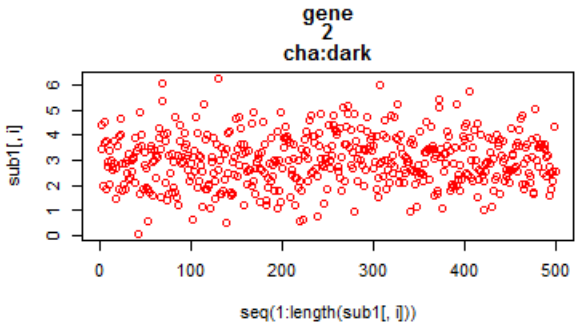
The first two genes are somewhat well-separated and so are the first three genes under different characters expressed.

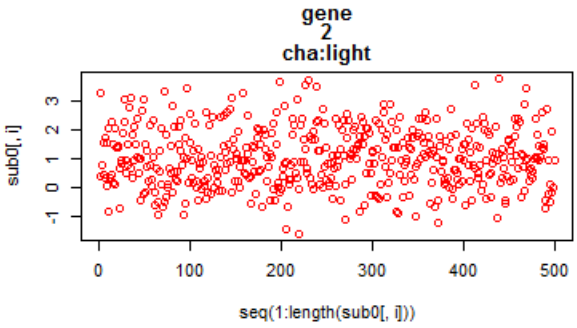
The result is obvious: The difference in the level of the expression of genes will result different character in human.

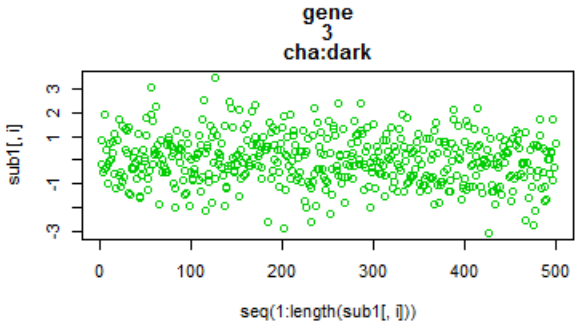
However, in the 3d-plot, we can find that one variable has the same levels in different groups.

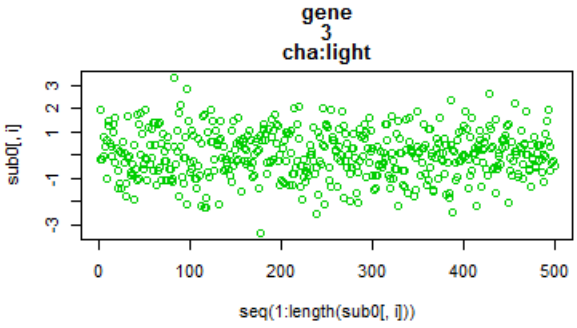

gene
1
cha:dark
sub1[, i]
seq(1:length(sub1[, i]))

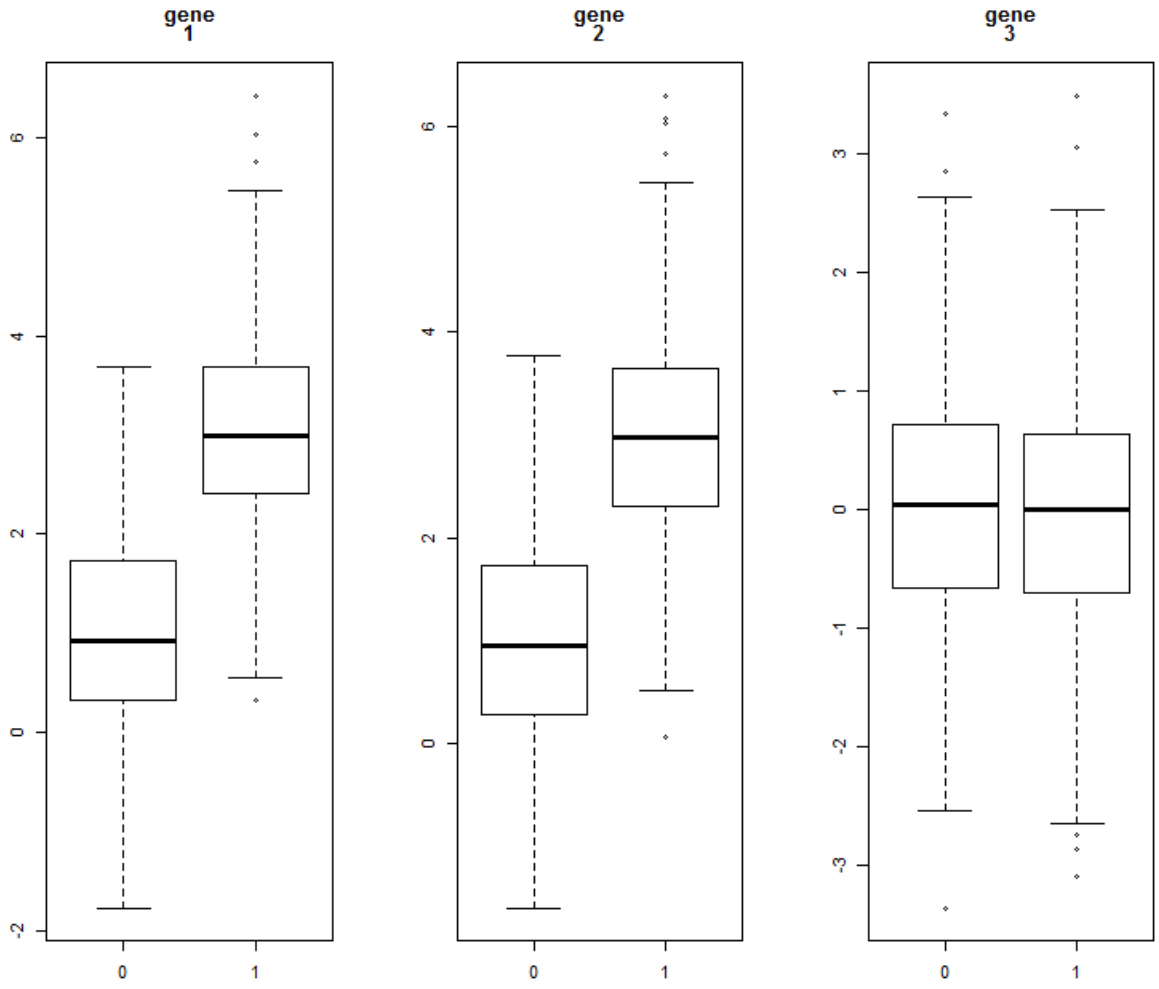

gene
1
cha:light
sub0[, i]
seq(1:length(sub0[, i]))


gene
2
cha:dark
sub1[, i]
seq(1:length(sub1[, i]))


gene
2
cha:light
sub0[, i]
seq(1:length(sub0[, i]))


gene
3
cha:dark
sub1[, i]
seq(1:length(sub1[, i]))


gene
3
cha:light
sub0[, i]
seq(1:length(sub0[, i]))

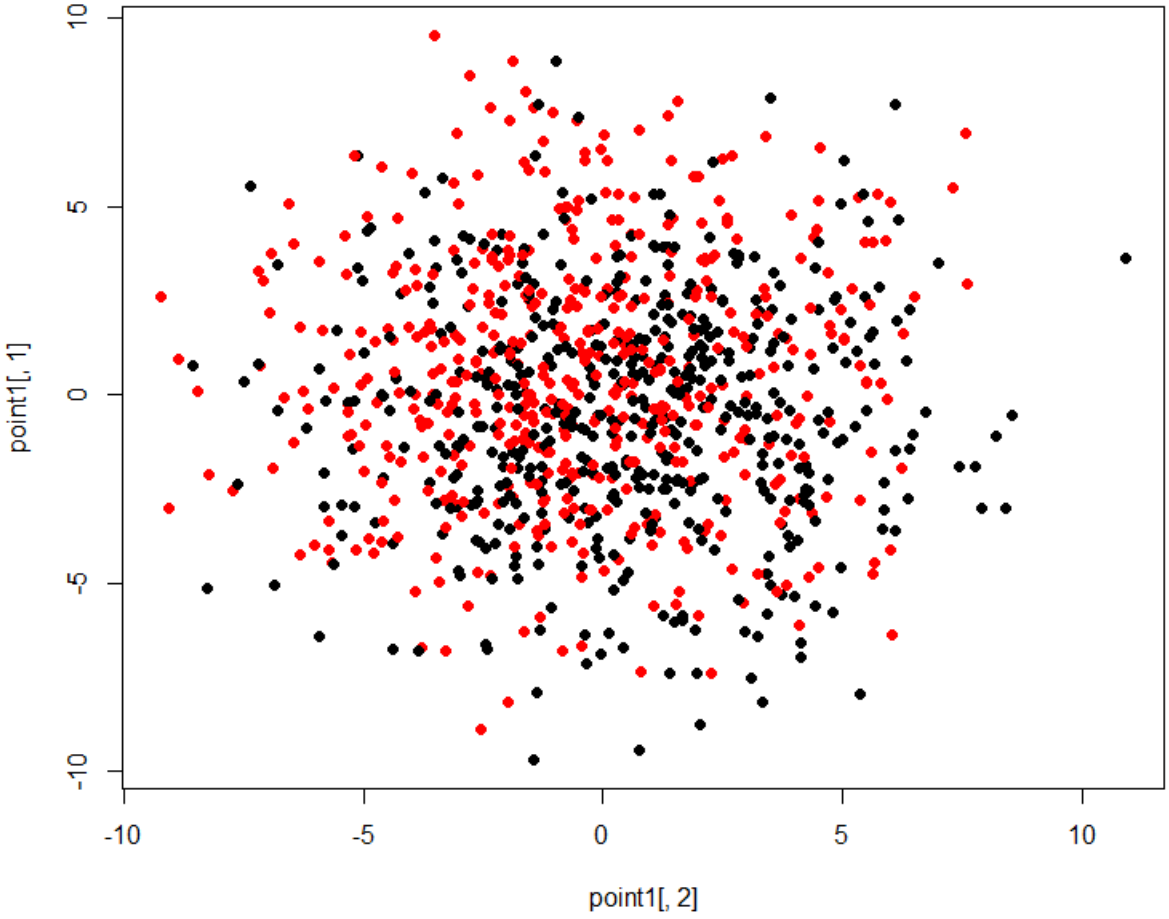


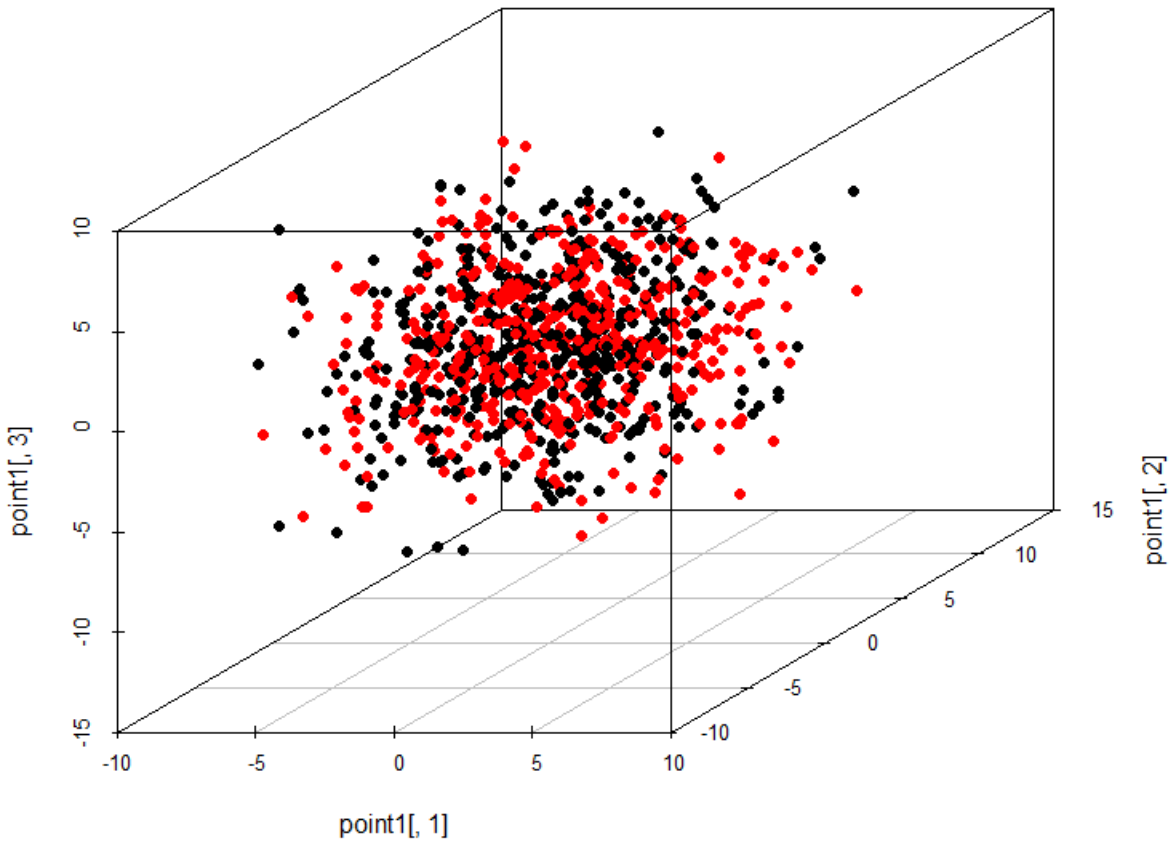
We can find the third gene has the same levels in different group, so it might not be a deterministic gene for eye's color.

(2)

Here, I want to have some tries to find a good similarity matrix:

First: Euclidean distance as similarity matrix.

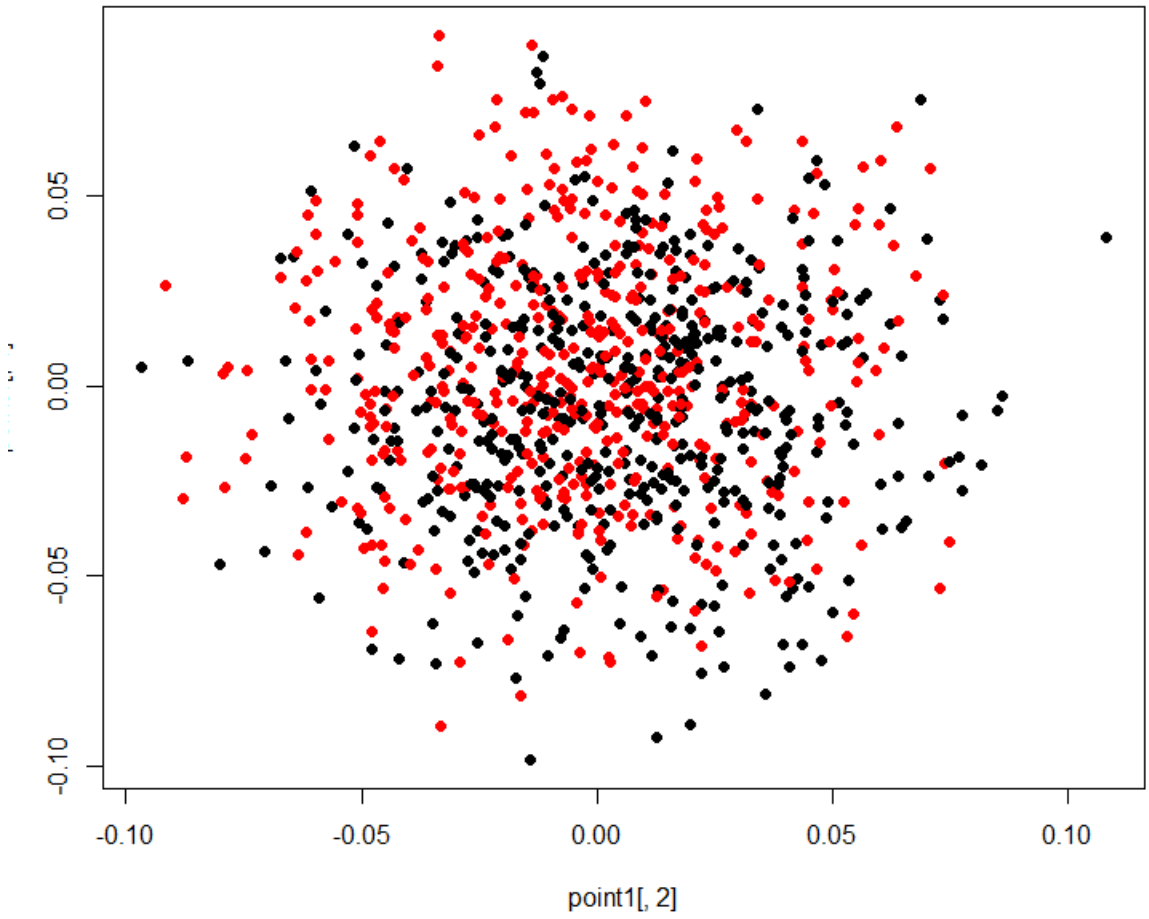

point1[, 1]
10
5
0
-5
-10
-10
-5
0
5
10
point1[, 2]

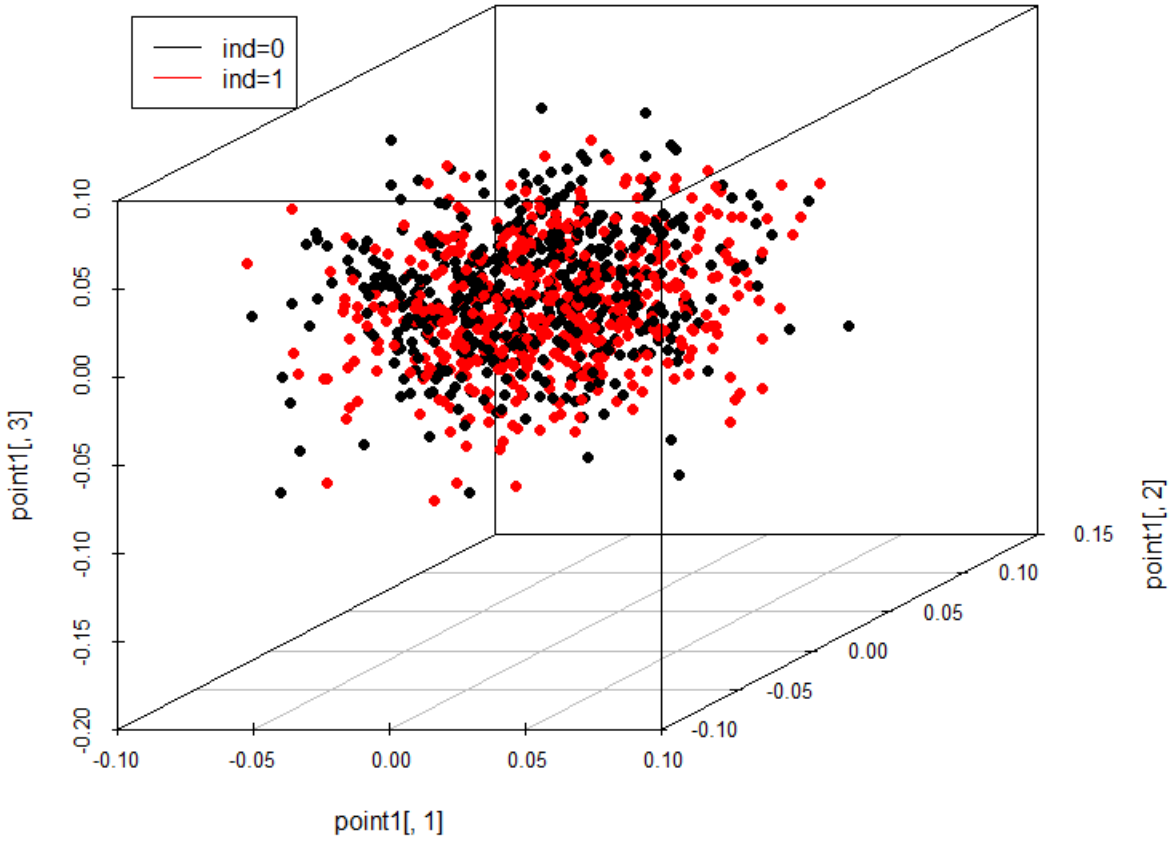


The result is very poor. (poor I mean not well-separated)

Then we try some other similarity matrix.

Correlation?


0.05
0.00
-0.05
-0.10
-0.10
-0.05
0.00
0.05
0.10
point1[, 2]

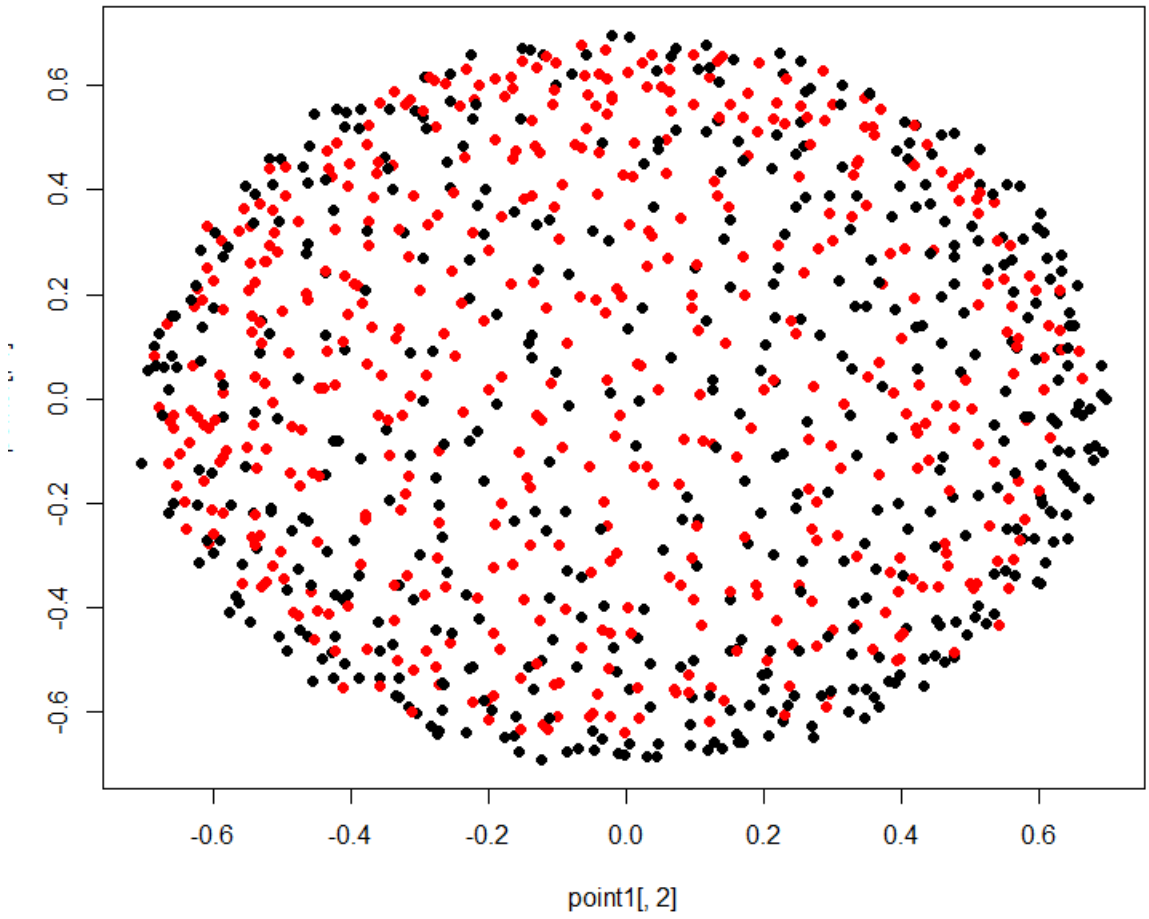


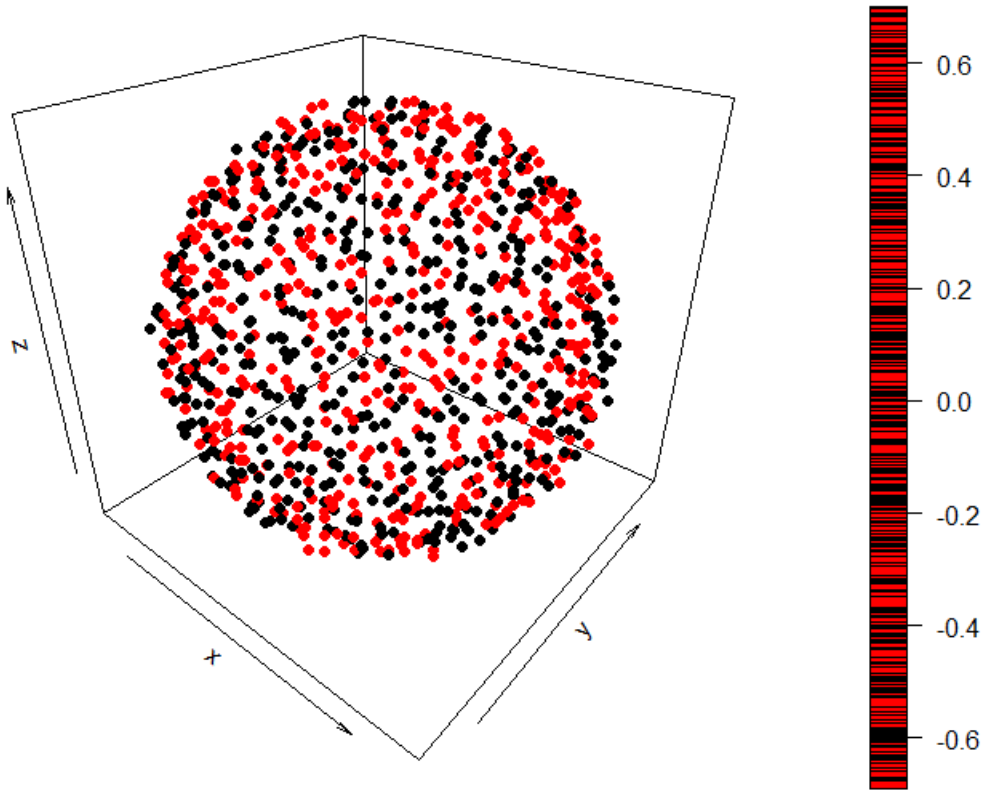
Not so good as well. It seems we cannot be so aimless. We should find some way to estimate a good dissimilarity or similarity matrix.

We start from Euclidean distance to perform non-metric MDS:

The result will not be shown because it is very poor.

We use the correlation matrix and convert it to dissimilarity matrix and then perform non-metric MDS.


0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
point1[, 2]



If we use non-metric MDS, which means we estimate dissimilarities by isometric regression, will have the result shown above. For 2-D, the recovery is not so bad, we can find the red and black point are mainly distributed in the circle or on the boundary of the circle. Well the 3D plot is not a good one, the black and red points are highly-mixed.

I don't think it is a good way to perform MDS on this data set. There are 5000 variables. We are not sure which one determine the color of eyes. What if there is only one or two particular genes determine the color? There will be so many noisy variables, which will result poor recovery.