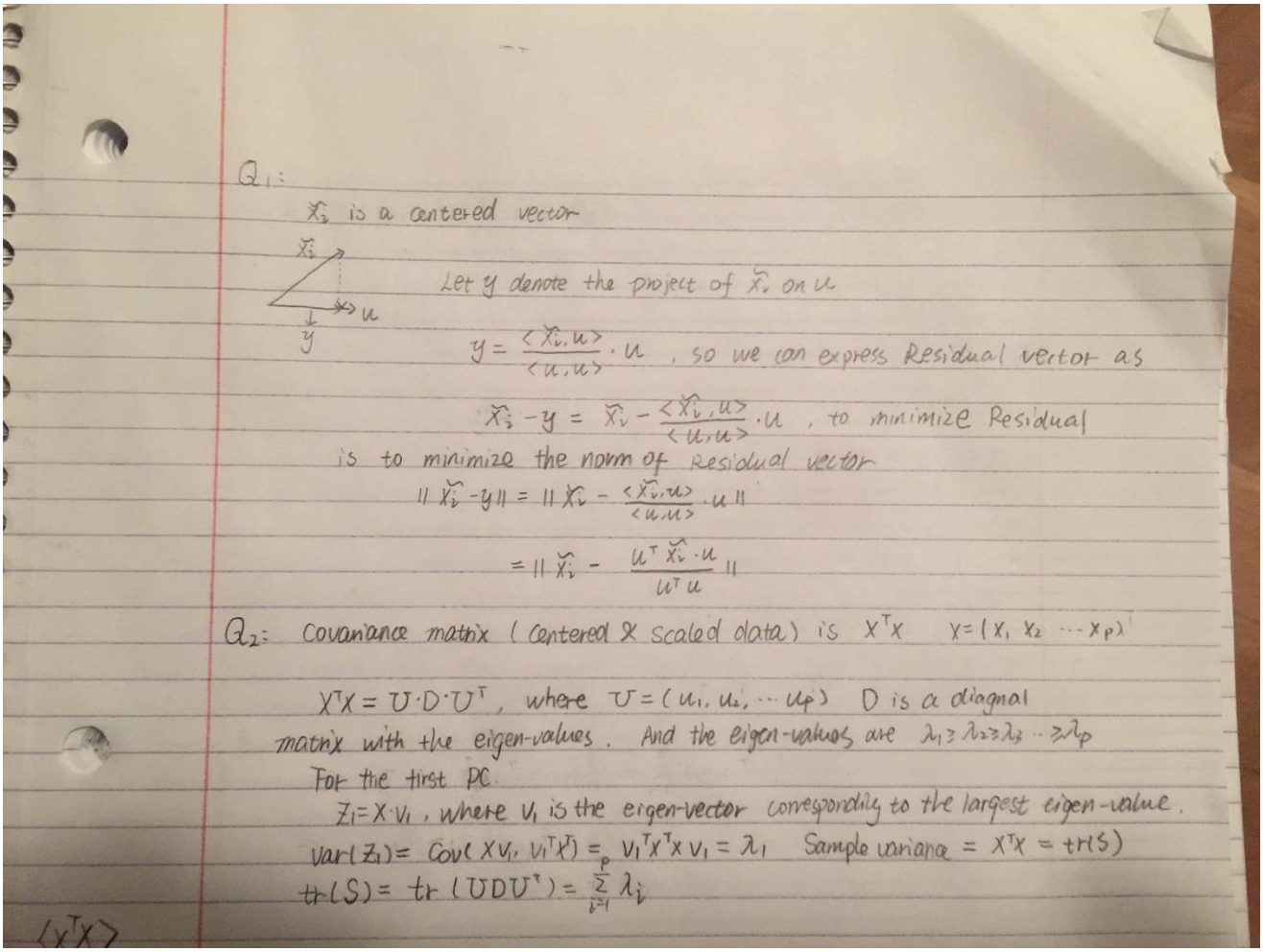
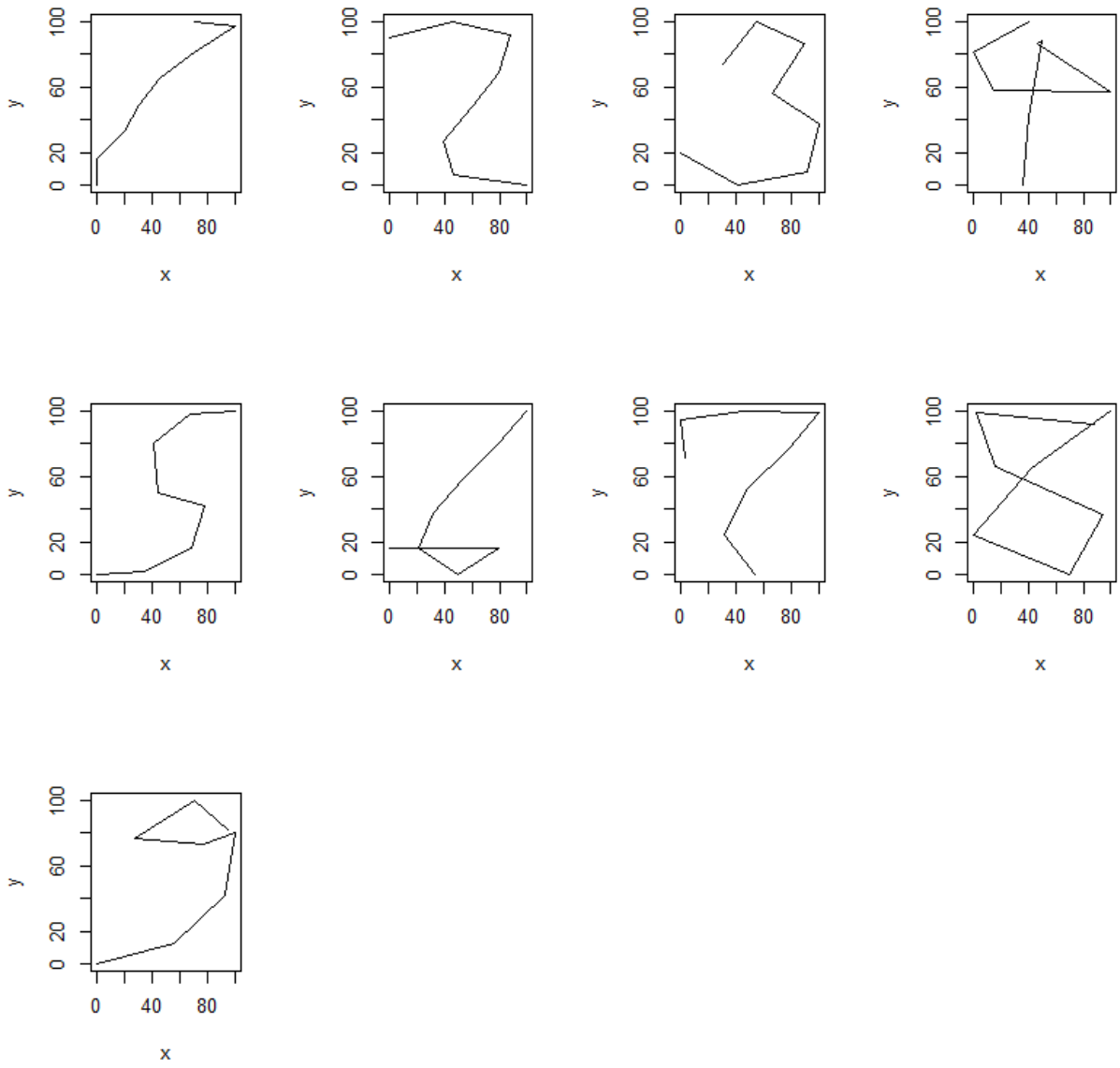


The first two problem's solution is shown above.

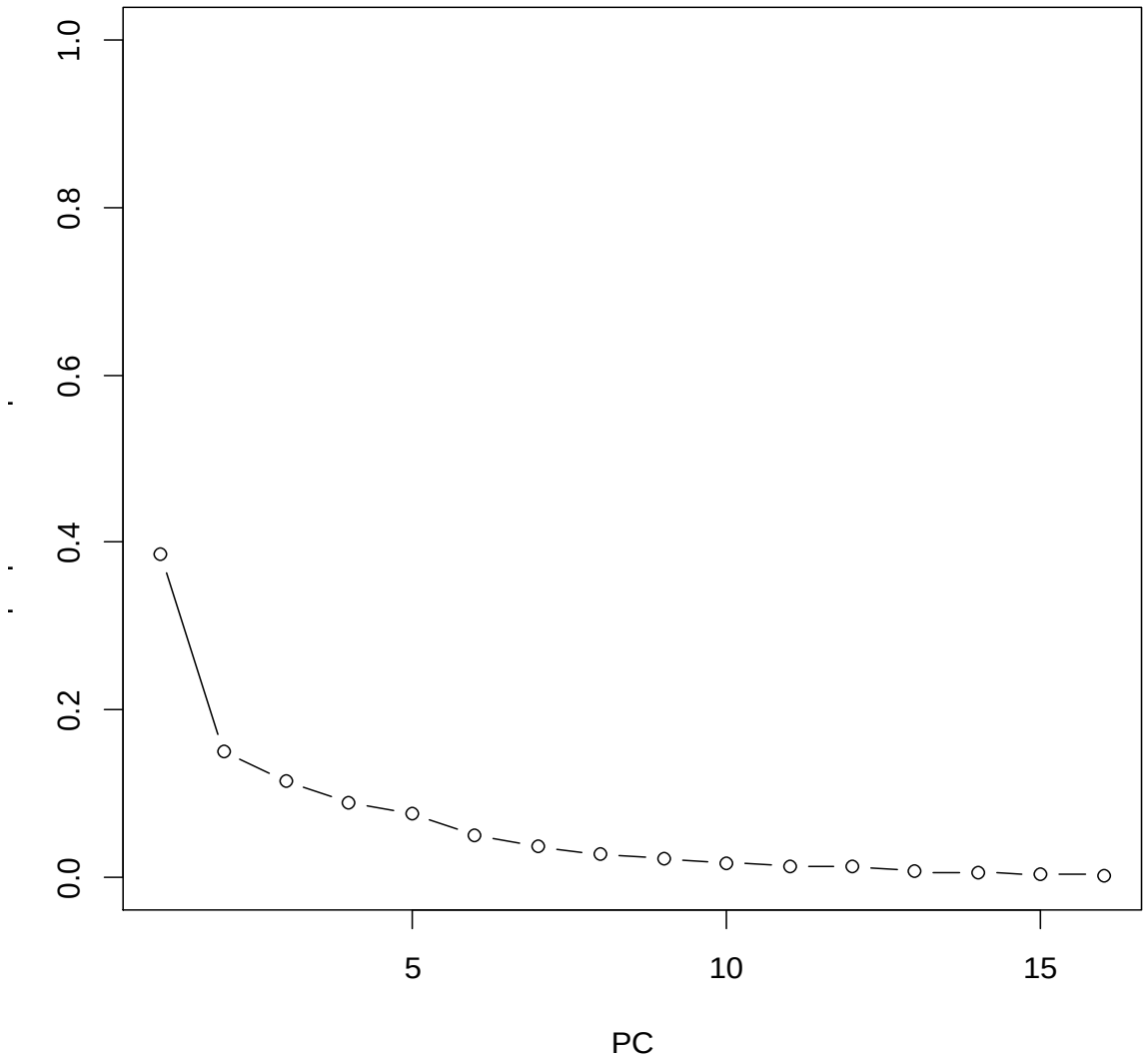
(1)

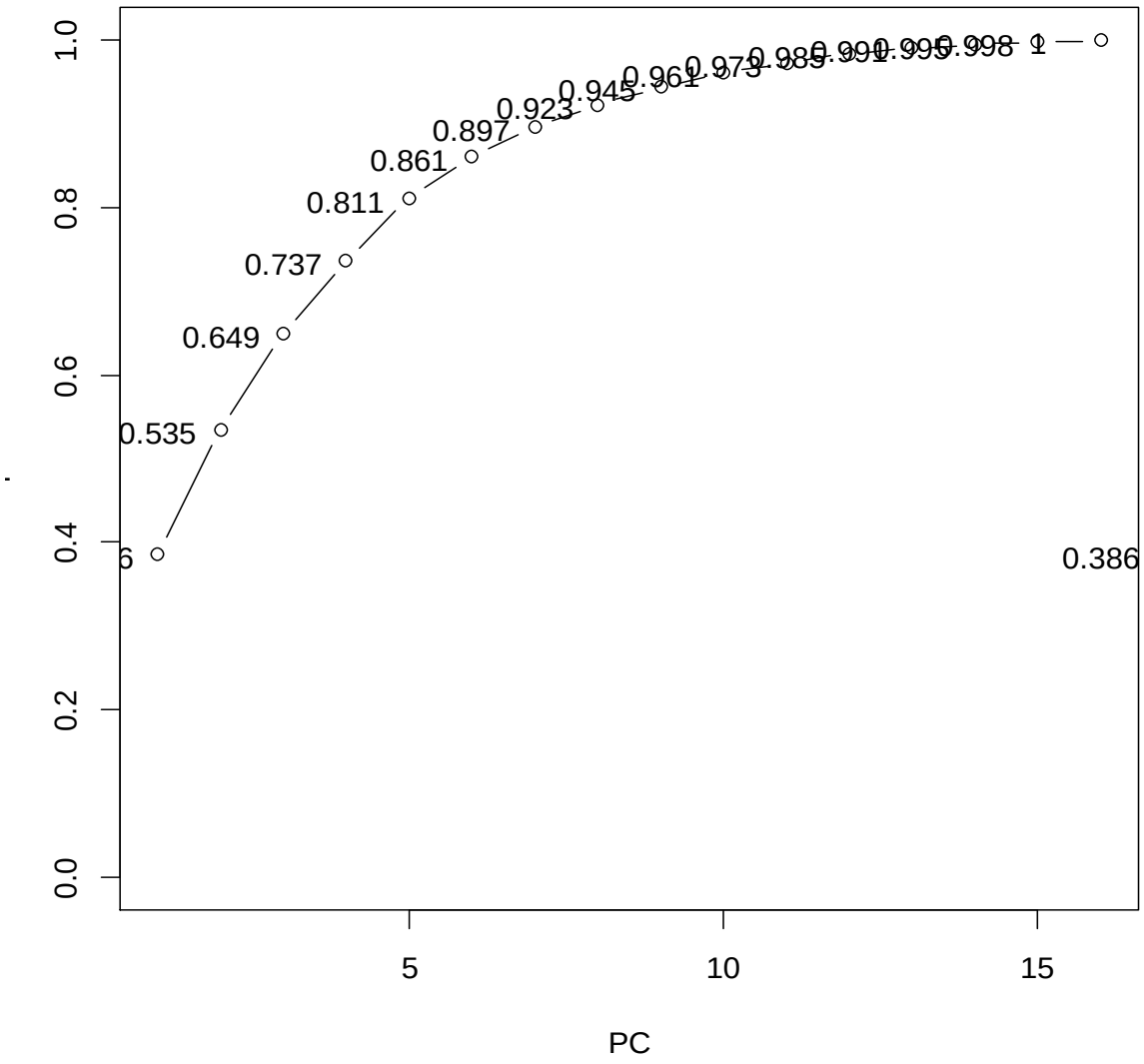


They are numbers!

（2）

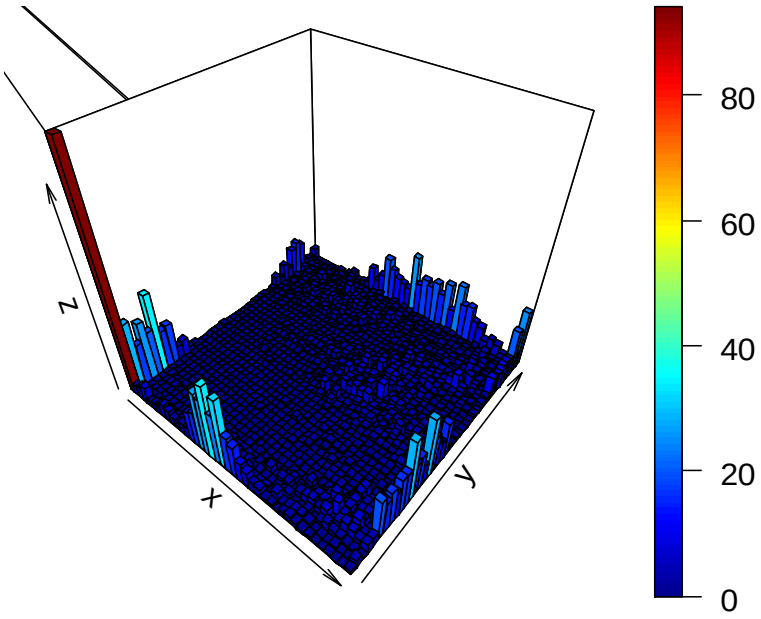
```
> pr.out$rotation
             PC1           PC2         PC3         PC4           PC5
V1  -0.27295715  0.275705343  0.05850097 -0.23947720  0.090625585
V2  -0.07797487 -0.056907632  0.64685212 -0.02752086 -0.064712900
V3  -0.29900237  0.157061826  0.34812961 -0.04346785  0.077074716
V4  -0.11806926  0.065020171  0.11377359  0.63541047 -0.395308871
V5  -0.02506130  0.508160362  0.00241109  0.34827641  0.124344435
V6   0.26960654  0.338210829 -0.24238700  0.21904609 -0.119868389
V7   0.07816379  0.468436317 -0.13212844 -0.19660122 -0.443012960
V8   0.32468924  0.262671099 -0.05807908 -0.17510910  0.094701254
V9  -0.23439340 -0.124345494 -0.25231579 -0.28743305 -0.360372606
V10  0.32215293 -0.022433329 -0.13374758 -0.05222543  0.346929694
V11  0.10464957 -0.382809918 -0.16981890  0.38620712 -0.057391576
V12  0.37342679  0.009399539  0.12072759 -0.01610819  0.195956090
V13  0.32997348 -0.184274595  0.13704660  0.01486819 -0.123396593
V14  0.29634544  0.058993745  0.29507119 -0.11684736 -0.213838774
V15  0.25808677 -0.150426776  0.05808749 -0.21162391 -0.487237293
V16 -0.24246512 -0.090262910 -0.36089435 -0.06515176 -0.007464463
> pr.sd
 [1] 6.16878827 2.38959885 1.82203898 1.40583413 1.19401539 0.79303538
 [7] 0.57966499 0.42238541 0.33767323 0.26534458 0.19761811 0.18383090
[13] 0.09499160 0.07281278 0.04497993 0.02738748
>
```


1.0
0.8
0.6
0.4
0.2
0.0
5
10
15
PC

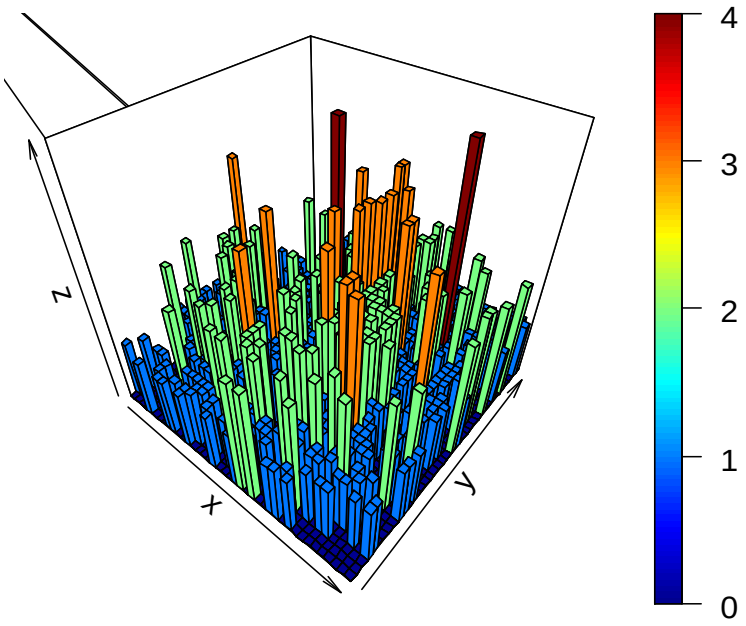


(3)

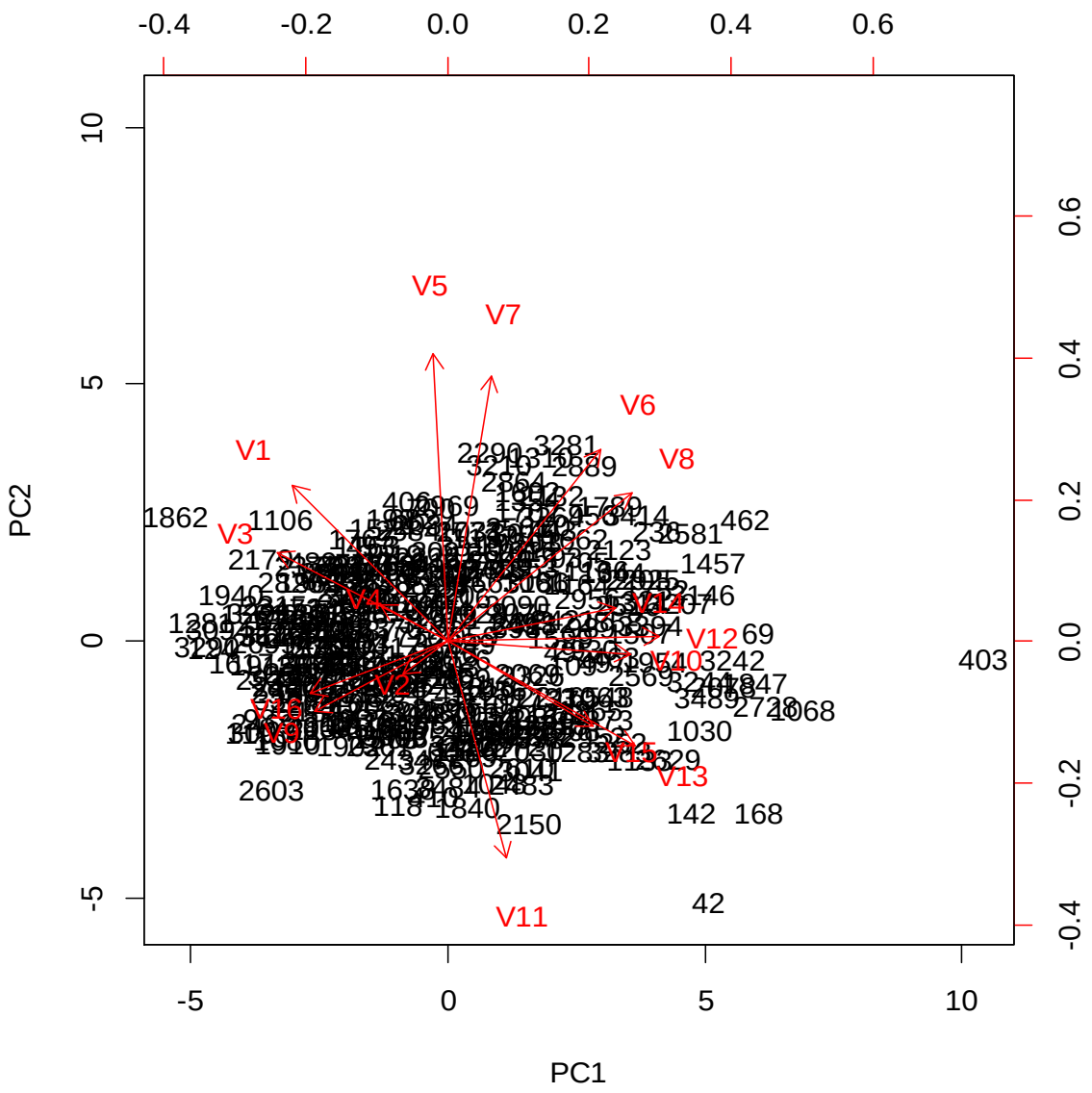
For the multivariate normal part. The 3D plot (x,y) against frequency is shown below.



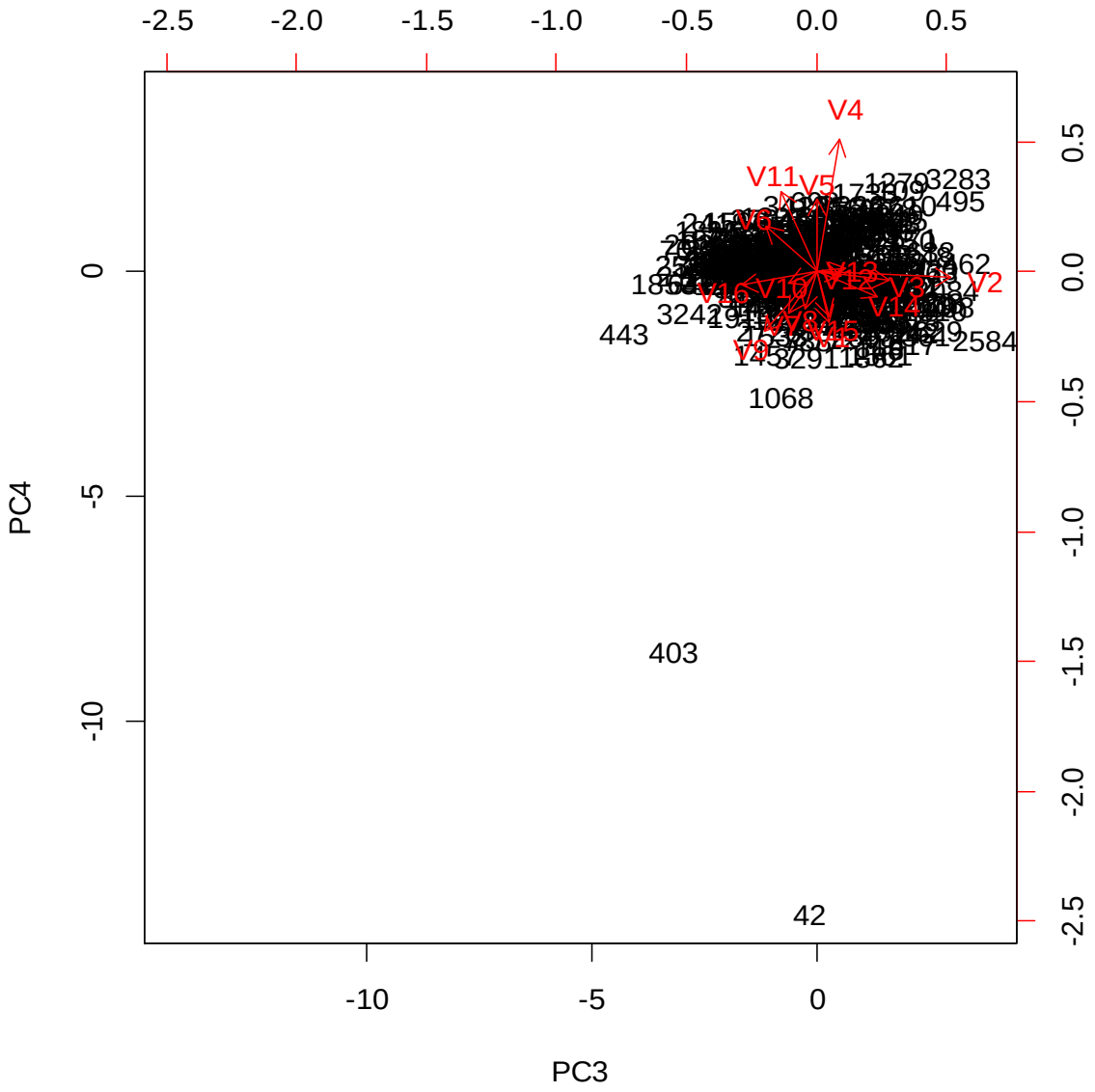
Then I sample from multivariate normal dist., with sample mean and sample variance, then plot the 3D histogram. When comparing these two plots, we can find that the sample does not obviously distribute as multivariate normal dist.



The first five PC contain almost 80% of the variation of the data. So, I would keep the first five PC.



The scores of each observation are shown above. We can find the first loading vector places much weight to v4 v2 v14 v12 v10 v3 v16 , and approximately equal weight to v1 v6 v8 v13 v15. These variables might contain the largest part of variance in the data. And the second loading vectors places much weight to the V5 V87 V11, and less weight to the others.



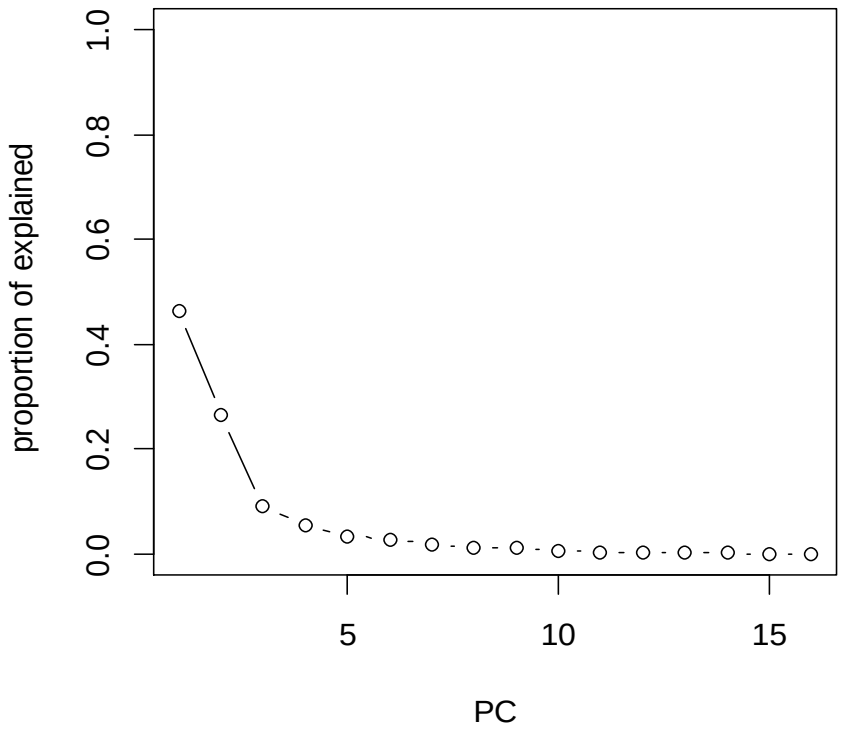
We can find the third loading vectors put much weight on v16 v10 v13 v3 v14 v2, and the fourth loading vectors show more weight on the others (these two vector have almost equal weight on v9 v8 v6)

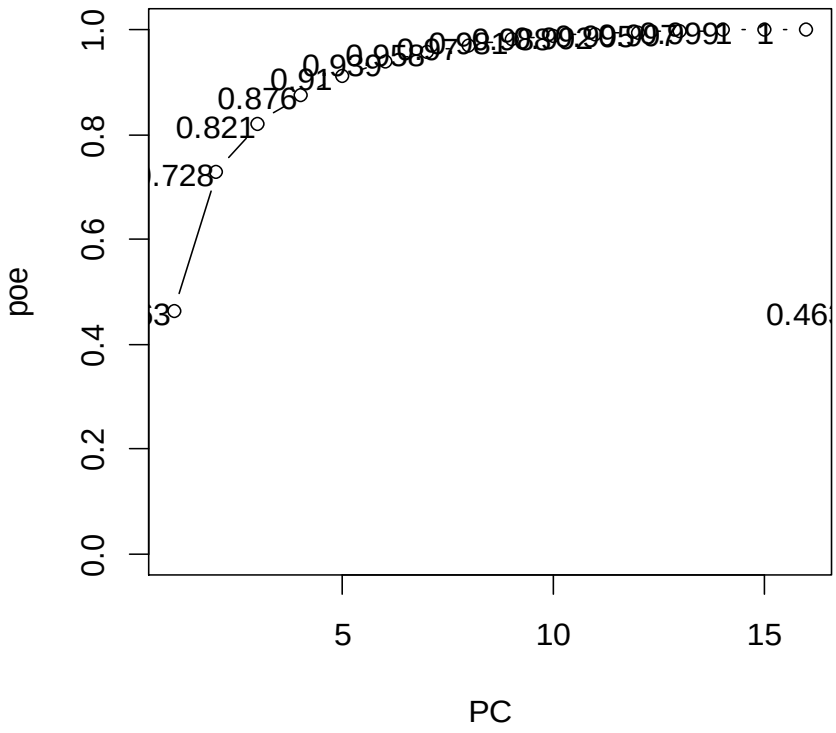
we can find some features of the tendency of variation on the data

(But notice that, for example, V1 and V2 seem to be a pair, their covariance might not simply the same as that of the covariance matrix, the covariance matrix only contain part of information of the data. So we cannot be sure that PCA does work here).

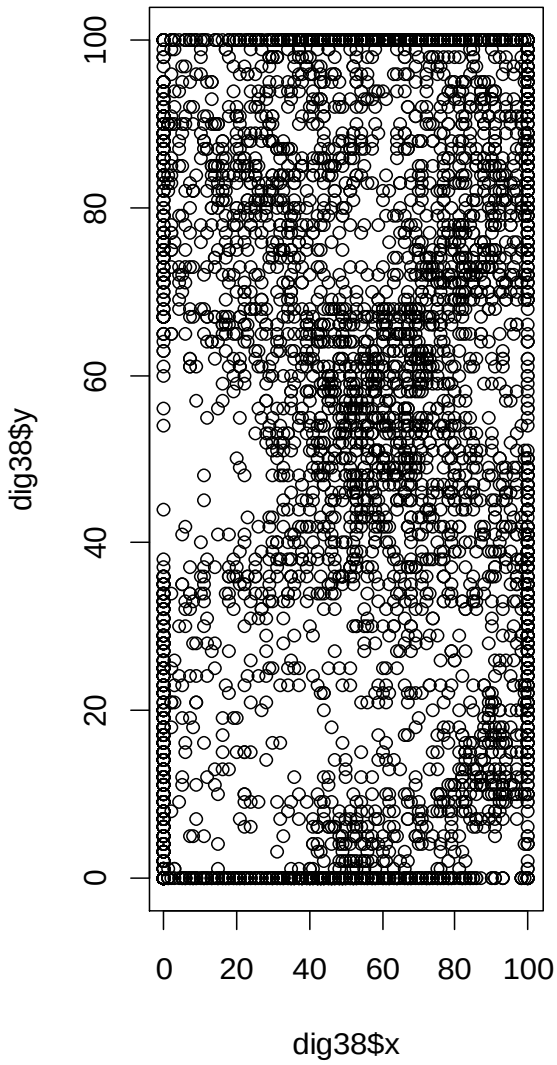
(5)

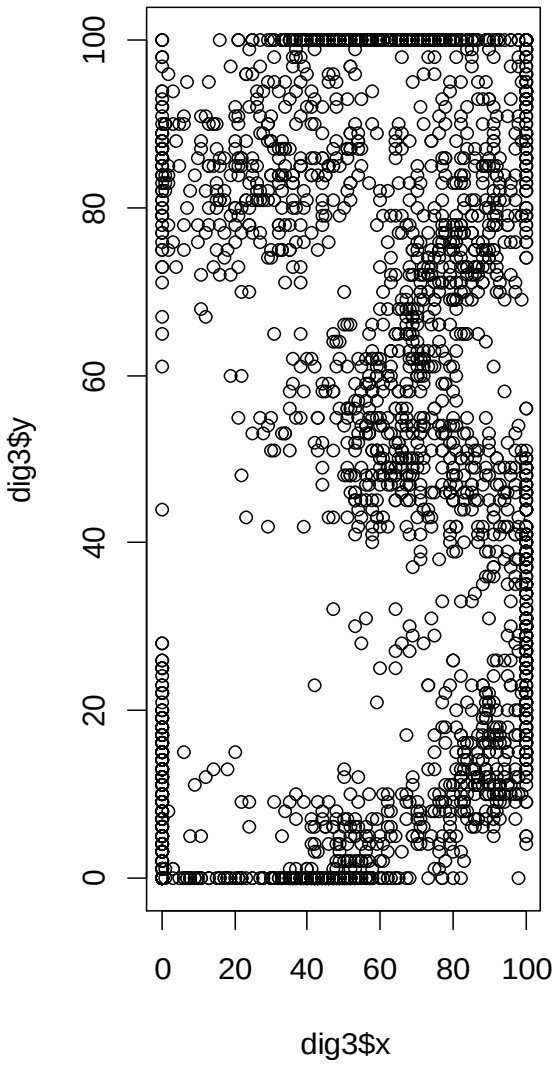
```
> pr.out2$rotation
            PC1         PC2           PC3           PC4           PC5
V1    0.1598237 -0.36034463 -0.153158961  0.2958060654 -0.084111113
V2    0.0816332 -0.06582932  0.631641326  0.4182388064  0.548540456
V3   -0.1666649  0.09490800 -0.497512783  0.5999591309  0.006545668
V4   -0.2429623 -0.32079810  0.089430926 -0.0154083560  0.215196866
V5   -0.2058076  0.26756929  0.235700179  0.4169463004 -0.411292328
V6   -0.3068475 -0.20897707  0.073132319 -0.1679959762 -0.001044127
V7   -0.1163443 -0.28752712  0.439128422 -0.0003598082 -0.630867206
V8   -0.3384576 -0.08962252  0.007155879 -0.2115497835  0.017981773
V9   -0.3039410 -0.07985938 -0.121582023 -0.1520042302  0.007759742
V10 -0.2361004  0.34637095  0.018213719 -0.1339437218  0.041490563
V11 -0.2285867  0.35262745  0.032235099 -0.0772313912  0.103536871
V12  0.2597963  0.31397035  0.076197169 -0.0876086921 -0.123048558
V13  0.2193356  0.29132260  0.166763034 -0.2075780429  0.064688887
V14  0.3554708  0.07207394 -0.026138423 -0.0190479771 -0.141973790
V15  0.2277606 -0.31565031 -0.132350119 -0.1799912069  0.090802268
V16  0.3495293 -0.09464686 -0.017265583 -0.0056488354 -0.134045270
> pr.sd2
 [1] 7.41090287 4.24488074 1.47673558 0.87852144 0.55474087 0.45924457
 [7] 0.29519431 0.20064279 0.18304242 0.10496023 0.06167136 0.04643434
[13] 0.03561043 0.02727335 0.01339920 0.00674549
```


proportion of explained
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
PC



It seems the first three PCs are enough to contain the information of the data. But as I said before, the covariance matrix does not contain all the information of the data (i.e. the data point appears in pair, not a single value!). We'd better check the scatterplot to make sure PCA works here.





As we can see, the first plot is just a mess, we cannot find any pattern of it. Thus we will not believe PCA works well in this data set. As for the other plot, there is an obvious tendency of how these data are arranged (in fact it is number 3), PCs could have a conclusion about the data tendency here.

Code：

```
getwd()

setwd("c:/users/an/desktop")

data=read.table("datahw2.txt",sep=",")

fix(data)
```

```
seg=split(data,data$V17)
attributes(seg)
seg3=seg$"3"
# PCA#
pr.out=prcomp(seg3[,-17],scale=T)
names(pr.out)
pr.out$rotation
pr.sd=pr.out$sdev^2
pr.var=sum(pr.sd)
evar=pr.sd/pr.var
evar
plot(evar,xlab="PC",ylab="proportion of explained",ylim=c(0,1),type='b')
plot(cumsum(evar),xlab="PC",ylab="poe",ylim=c(0,1),type='b')
text(seq(0,16),cumsum(evar),labels=round(cumsum(evar),3))
biplot(pr.out,scale=0)
xy=cbind(seg3$V1,seg3$V2)
feq=count(seg3,c("V1","V2"))
names(feq)

biplot(pr.out,choices=1:2,scale=0)
biplot(pr.out,choices=3:4,scale=0)
seg8=seg$"8"
seg38=rbind(seg3,seg8)
pr.out2=prcomp(seg3[,-17],scale=T)
names(pr.out2)
pr.out2$rotation
pr.sd2=pr.out2$sdev^2
pr.var2=sum(pr.sd2)
evar2=pr.sd2/pr.var2
```

```
evar2
# problem 1#
plot(evar2,xlab="PC",ylab="proportion of explained",ylim=c(0,1),type='b')
plot(cumsum(evar2),xlab="PC",ylab="poe",ylim=c(0,1),type='b')
text(seq(0,16),cumsum(evar2),labels=round(cumsum(evar2),3))
par(mfrow=c(3,4))
for (i in 1:length(unique(data$V17))){
for (k in 1:dim(data)[1]){
if (data[k,17]==i){
x=NULL;
y=NULL;
for (m in 0:7){
x=c(x,data[k,2*m+1]);
y=c(y,data[k,2*m+2]);
}
plot(y~x,type='l');break;
}
}
}
# normality test#
dig3=NULL;
dig3$x=c(seg3$V1,seg3$V3,seg3$V5,seg3$V7,seg3$V9,seg3$V11,seg3$V13,seg3$V15)
dig3$y=c(seg3$V2,seg3$V4,seg3$V6,seg3$V8,seg3$V10,seg3$V12,seg3$V14,seg3$V16)
x=cut(dig3$x,40)
y=cut(dig3$y,40)
c=table(x,y)
hist3D(z=z,border="black")
# sample from the mutivariate with mean and covariance matrix#
mean(dig3$x)
```

```
mean(dig3$y)
dig3.frame=as.data.frame(dig3)
cov(dig3.frame)
library(MASS)
rmv=mvrnorm(1000,mu=c(mean(dig3$x),mean(dig3$y)),Sigma=cov(dig3.frame))
rmv.s=rmv.frame[rmv.frame$x>=0&rmv.frame$x<=100&rmv.frame$y>=0&rmv.frame$y<=100,]
rm.x=cut(rmv.s$x,40)
rm.y=cut(rmv.s$y,40)
p=table(rm.x,rm.y)
hist3D(z=p,border="black")
#last question#
dig38=NULL;
dig38$x=c(seg38$V1,seg38$V3,seg38$V5,seg38$V7,seg38$V9,seg38$V11,seg38$V13,seg38$V15)
dig38$y=c(seg38$V2,seg38$V4,seg38$V6,seg38$V8,seg38$V10,seg38$V12,seg38$V14,seg38$V16)
par(mfrow=c(1,2))
plot(dig38$y~dig38$x)
plot(dig3$y~dig3$x)
```