W4415 Multivariate Statistical Inference Homework 2

Q1

The best first sample principal component

$$\boldsymbol{u} = arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n} \left\| \tilde{x}_i - \frac{\boldsymbol{u^T}\tilde{x}_i}{\boldsymbol{u^T u}}\boldsymbol{u} \right\| = arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n}\sum_{j=1}^{q} \left(\tilde{x}_{ij} - u_j\tilde{x}_{ij}u_j\right)^2 =$$

$$arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n}\sum_{j=1}^{q} (\tilde{x}_{ij} - u_j x_i^T \boldsymbol{u})^2 = arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^T x_i - 2\boldsymbol{u^T} x_i^T x_i \boldsymbol{u} + (\boldsymbol{u^T} x_i^T \boldsymbol{u})^{\boldsymbol{T}} (\boldsymbol{u^T} x_i^T \boldsymbol{u}))$$

$$= arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^T x_i - \boldsymbol{u^T} x_i^T x_i \boldsymbol{u}) = arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} \boldsymbol{u^T} x_i^T x_i \boldsymbol{u} = arg\max_{u:\|u\|_2=1} \frac{\sum_{i=1}^{n} \frac{1}{n}(\boldsymbol{u} x_i^T x_i \boldsymbol{u})}{\boldsymbol{u^T u}}$$

where $\frac{1}{n}(\boldsymbol{u} x_i^T x_i \boldsymbol{u})$ is the sample covariance.

Q2

The diagonal term of a covariance matrix is the variance for each element, so this is equivalent to the trace of the covariance matrix, aka the sum of the eigenvalues.

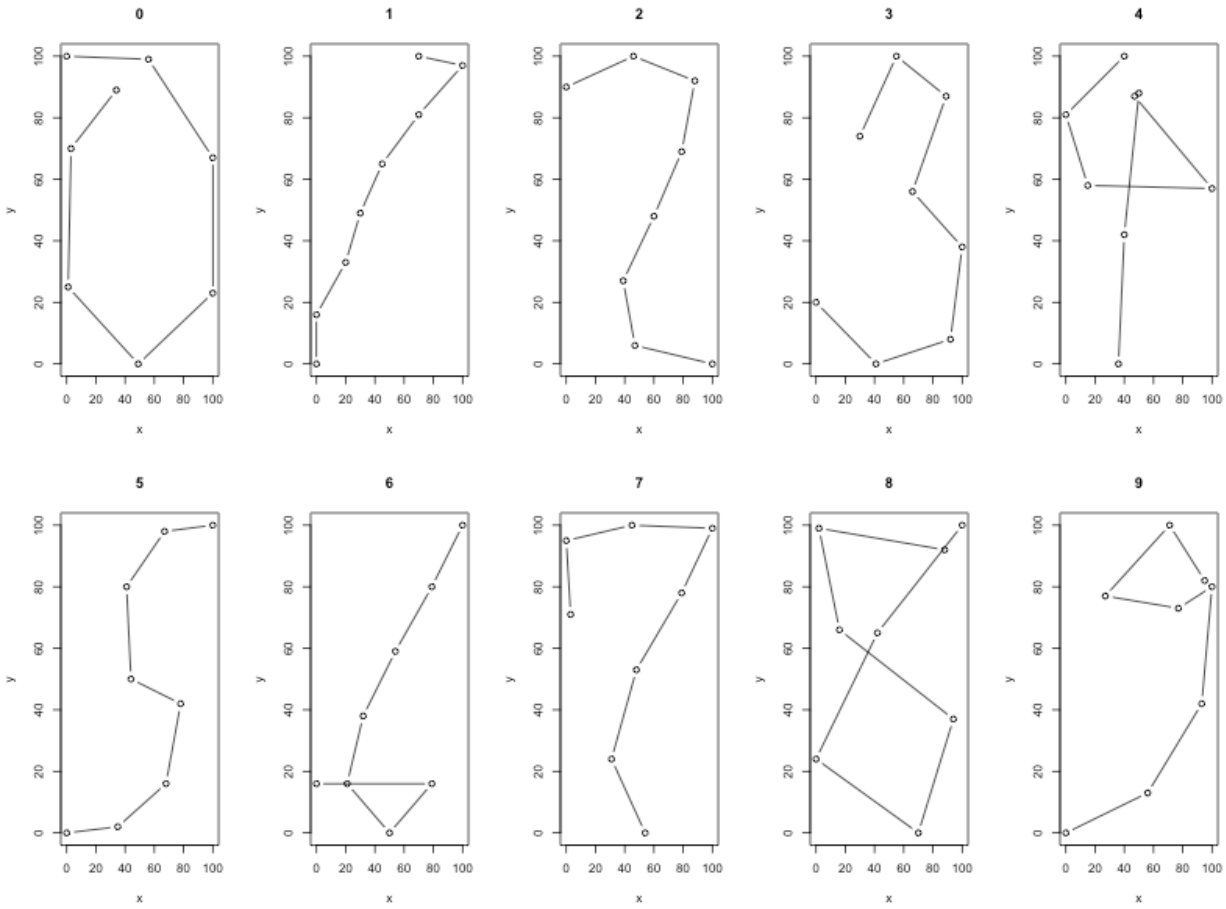
Assume U as eigenvectors, and $UU^T = I$.

$< x^T x >= C_x$ =trace$(C_x) = trace(UU^T C_x) = trace(U^T C_x U) = trace(\Lambda) = \sum_i \lambda_i$

Q3

(1)

```
setwd("~/Learning/Multivariate Statistical Inference/")
datahw2 <- read.csv("datahw2.txt", header=FALSE)
par(mfrow=c(2,5))
for(i in 0:9){
  index=which(datahw2$V17==i)[1]
  x=as.numeric(datahw2[index,c(1,3,5,7,9,11,13,15)])
  y=as.numeric(datahw2[index,c(2,4,6,8,10,12,14,16)])
  plot(x,y,type="b",main=i)
}
```



For each digit, its movement is to draw the digit. For example, the visualization of first observation for digit 0 looks like a "0".

(2)

```
library(car)
pendigit3=datahw2[which(datahw2$V17==3),]
head(pendigit3)
```

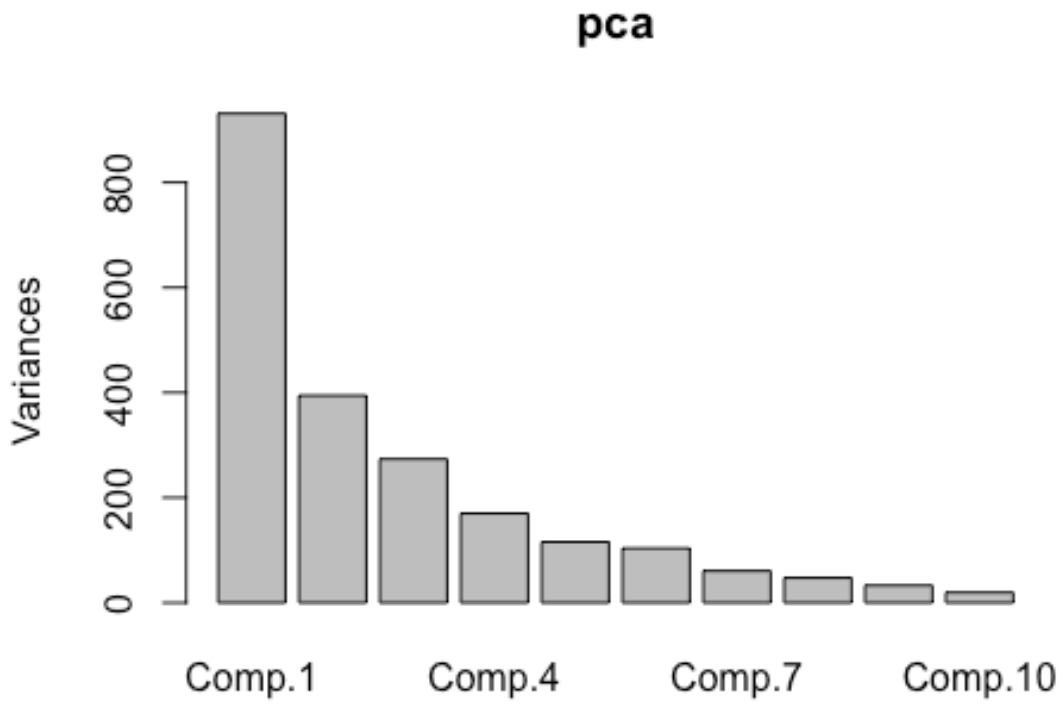
```
##    V1 V2 V3  V4  V5  V6 V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17
## 17 30 74 55 100  89  87 66  56 100  38  92   8  41   0   0  20   3
## 32 41 84 73 100 100  82 62  60  97  38  91   8  42   0   0  19   3
## 42 59 89 42  23  29  42 25 100 100  82  75  46  98   0   0   3   3
## 69  0 76 29  95  92 100 81  77  75  56 100  35  85  13  31   0   3
## 70 38 65 36  98 100 100 99  69  62  55  96  26  55   0   0  15   3
## 72  0 82 34 100  78  90 49  62  80  49 100  18  62   0  18  13   3
```

```
pca3=princomp(pendigit3[,1:16])
summary(pca3)
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3      Comp.4
## Standard deviation     30.5143901 19.8618329 16.5382037 13.03719789
## Proportion of Variance  0.4229384  0.1791869  0.1242351  0.07720337
## Cumulative Proportion   0.4229384  0.6021253  0.7263605  0.80356382
##                             Comp.5      Comp.6     Comp.7     Comp.8
```

```
## Standard deviation     10.75826624 10.18506982 7.78149149 6.88859781
## Proportion of Variance  0.05257173  0.04711896 0.02750384 0.02155407
## Cumulative Proportion   0.85613555  0.90325452 0.93075836 0.95231243
##                            Comp.9     Comp.10     Comp.11     Comp.12
## Standard deviation     5.74284846 4.424042087 4.221239041 3.851882078
## Proportion of Variance 0.01498037 0.008890091 0.008093709 0.006739283
## Cumulative Proportion  0.96729280 0.976182892 0.984276601 0.991015884
##                           Comp.13     Comp.14     Comp.15      Comp.16
## Standard deviation     2.9331367 2.460930392 1.840622081 1.3159754128
## Proportion of Variance 0.0039078 0.002750847 0.001538852 0.0007866169
## Cumulative Proportion  0.9949237 0.997674531 0.999213383 1.0000000000
```

```
screeplot(pca)
```



The principal components and their associated variances

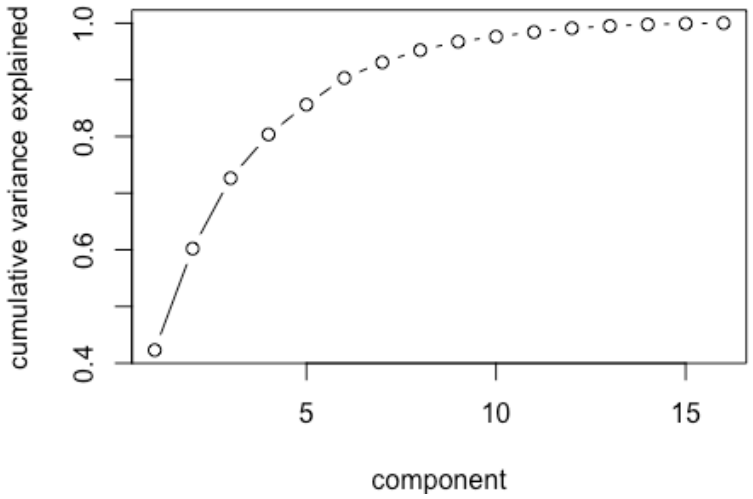
```
(pca$sdev)^2
```

```
##  Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 931.128005 394.492406 273.51218 169.968529 115.74029 103.735647 60.551610
##  Comp.8     Comp.9    Comp.10    Comp.11    Comp.12    Comp.13    Comp.14
## 47.452780  32.980308  19.572148  17.818859  14.836996   8.603291 6.056178
##  Comp.15    Comp.16
## 3.387890   1.731791
```

```
cumsum((pca3$sdev)^2)/sum((pca3$sdev)^2)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 0.4229384 0.6021253 0.7263605 0.8035638 0.8561356 0.9032545 0.9307584
##    Comp.8    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14
## 0.9523124 0.9672928 0.9761829 0.9842766 0.9910159 0.9949237 0.9976745
##   Comp.15   Comp.16
## 0.9992134 1.0000000
```

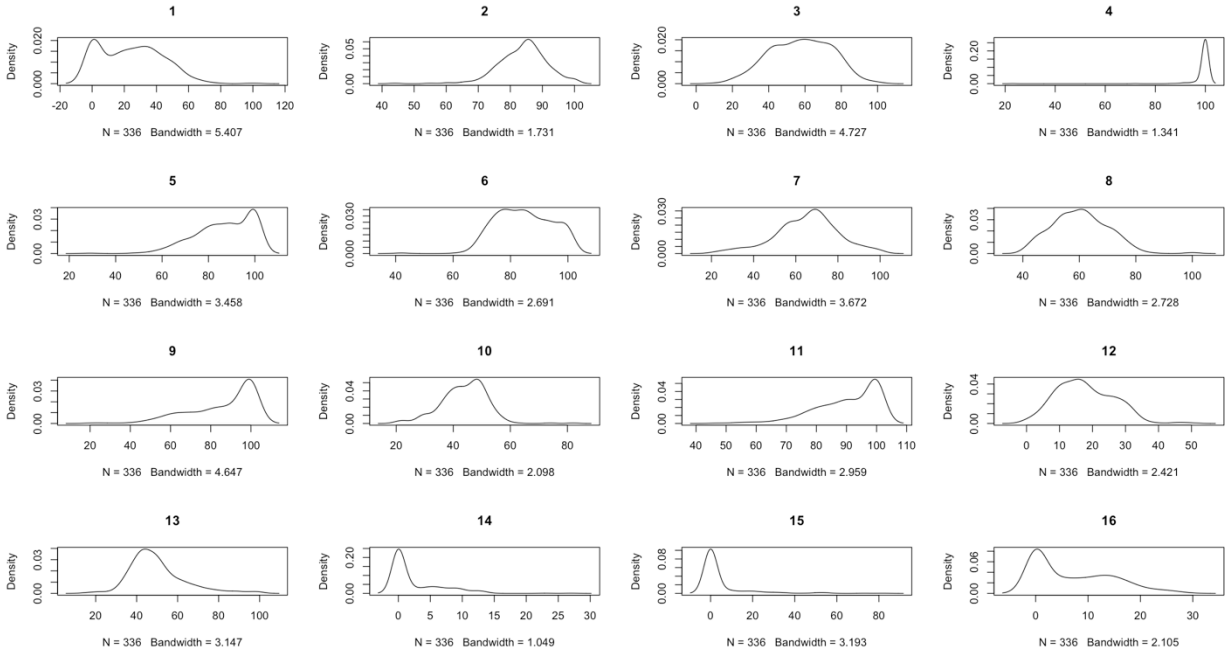
```
plot(seq(1:16),cumsum((pca$sdev)^2)/sum((pca$sdev)^2),type="b",xlab="componen
t",ylab="cumulative variance explained")
```



(3)

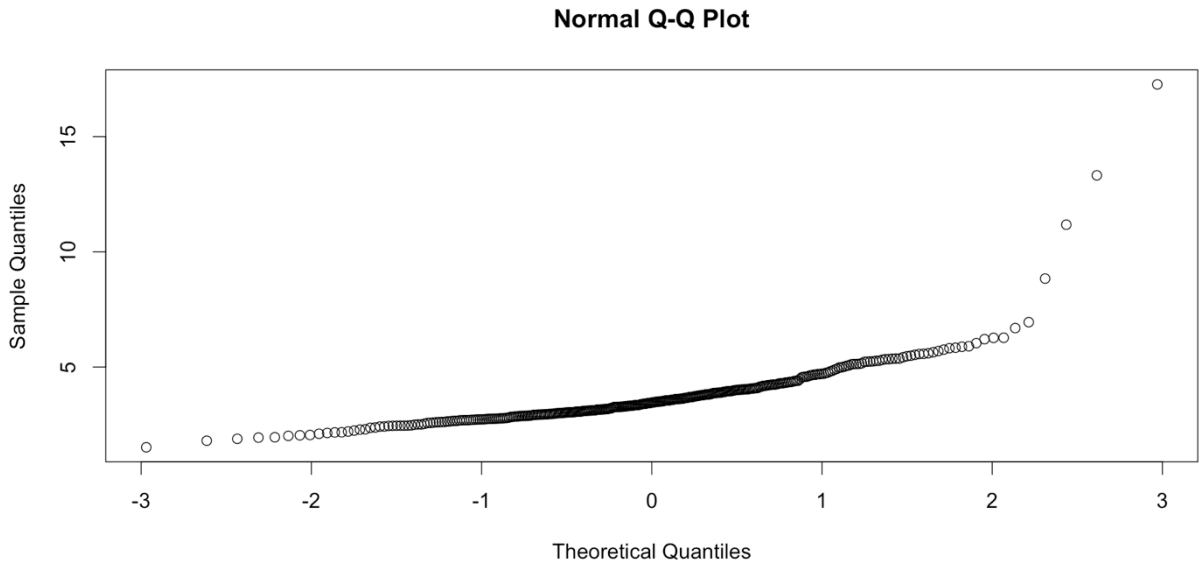
From the density plots matrices shown below, the data of (x_i, y_i) do not look like they are from a Multivariate Normal Distribution. For example, the distribution of V4 is centered at the extreme value 100, while for V16, its mass is near 0.

```
for(i in 1:16){
  plot(density(pendigit3[,i]),main = i)
}
```

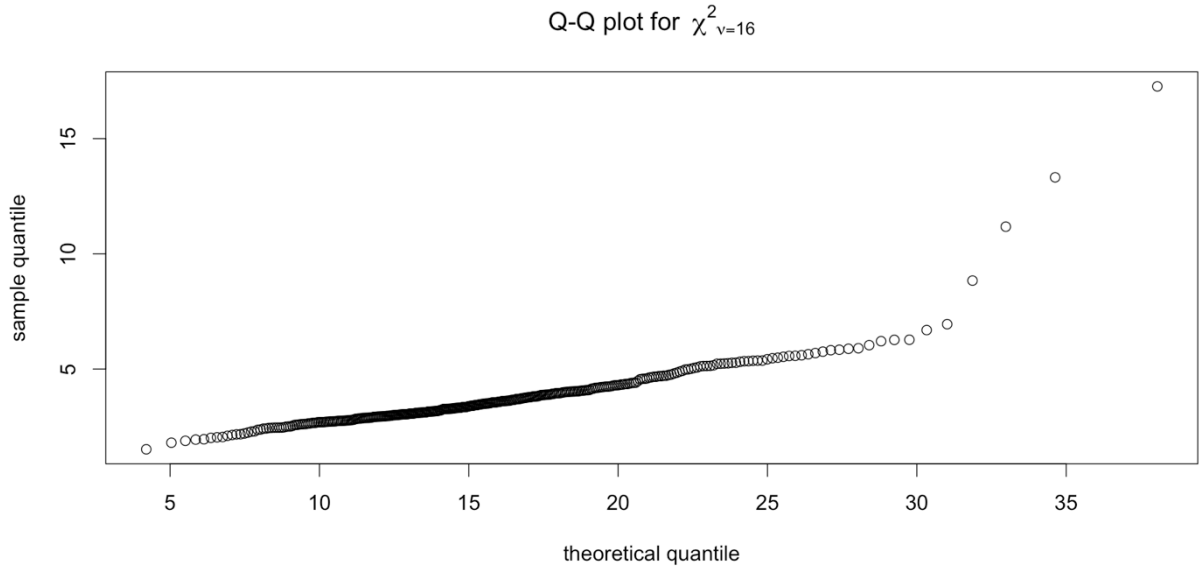



```
##############################
# Code block from hw01.R solutions
MD <- function(X){
  mu <- apply(X, 2, mean)
```

```
  sigma <- cov(X)
  md <- apply(X, 1, function(x) sqrt(t(x - mu) %*% solve(sigma) %*% (x
- mu)))
  return(md)
}
x<-MD(pendigit3[,1:16])
qqnorm(x)
```

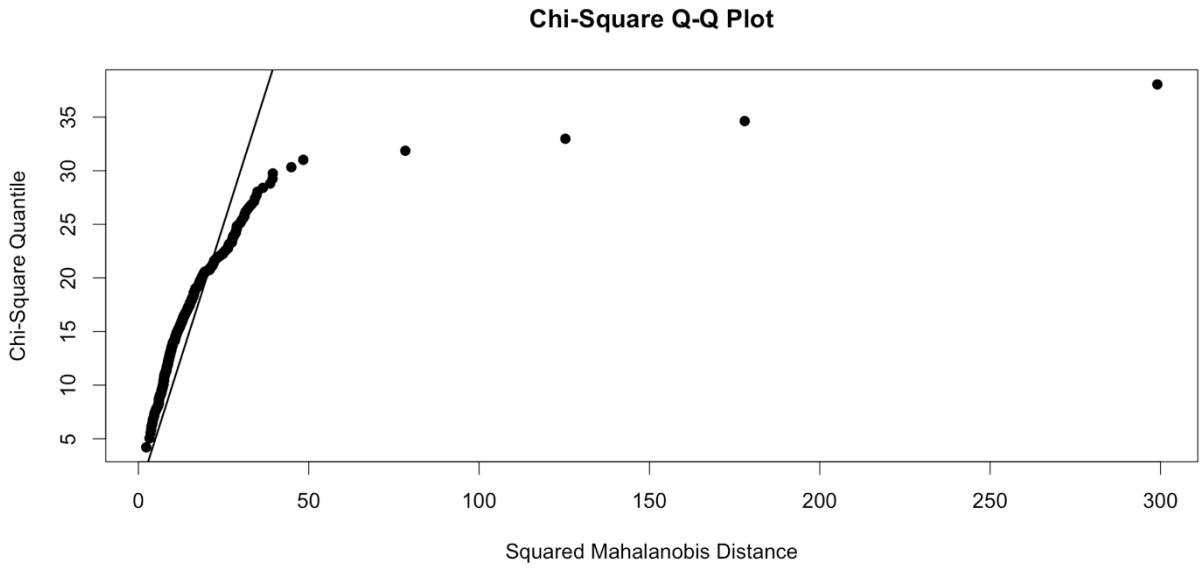



```
qqplot(qchisq(ppoints(length(x)), df = 16), x,
       main = expression("Q-Q plot for" ~~ {chi^2}[nu == 16]),
       xlab = "theoretical quantile", ylab = "sample quantile")
```




```
library(MVN)
hzTest(pendigit3[,1:16],cov=TRUE,qqplot=TRUE)
```

```
## Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : pendigit3[, 1:16]
##
##  HZ      : 1.356549
##  p-value : 0
##
##  Result  : Data are not multivariate normal.
## ---------------------------------------------
```

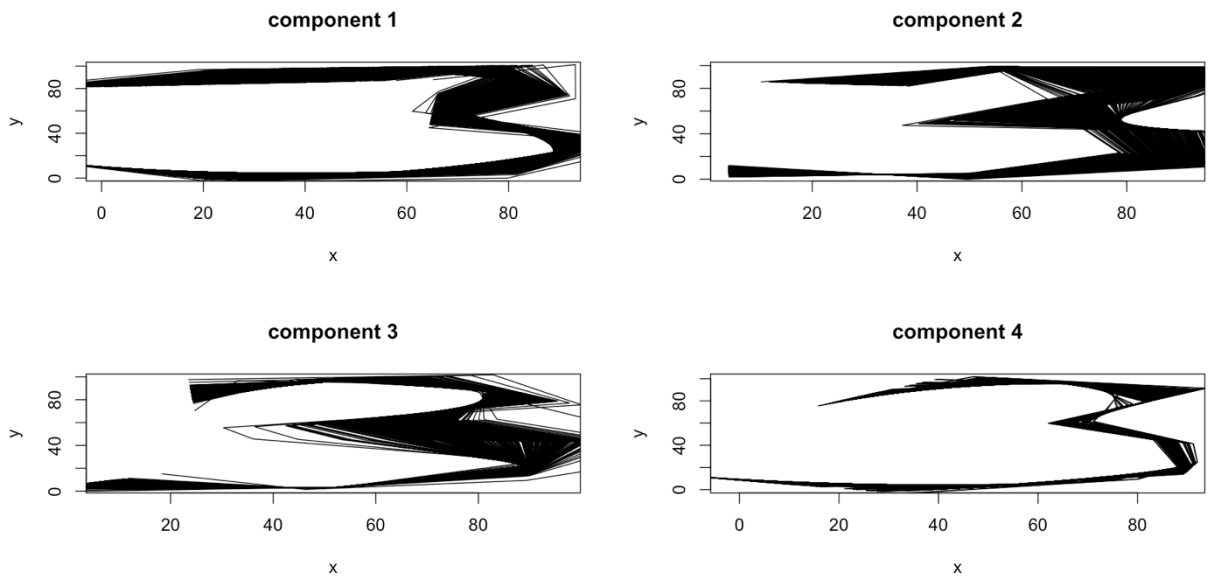


From the qqnorm, qqplot and Henze-Zirkler's Multivariate Normality Test shown above, the original data are obviously not Multivariate Normal Distribution as the quantile is very far from normal.

I would like to keep the first 8 components as they contain 95% of the original information which could be seen from the second line of page 6.

(4)

```
for(j in 1:4){
  w<-t(pca$loadings[,j])
  y1<-w%*%t(scale(pendigit3[,1:16],center=TRUE,scale=FALSE))
  proj.data1<-t(y1)%*%w+
matrix(rep(colMeans(pendigit3[,1:16]),nrow(pendigit3)),ncol=16,byrow=T)
  x0=as.numeric(proj.data1[1,c(1,3,5,7,9,11,13,15)])
  y0=as.numeric(proj.data1[1,c(2,4,6,8,10,12,14,16)])
  plot(x0,y0,type="l",main=sprintf("component %d",j),xlab="x",ylab="y")
  for(i in 2:336){
      lines(as.numeric(proj.data1[i,c(1,3,5,7,9,11,13,15)]),as.numeric(proj.d
ata1[i,c(2,4,6,8,10,12,14,16)]))
  }
  proj.data1=vector()
  i=2
}
```



The feature of digit 3 is preserved by each of the principal component.

(5)

```
pendigit38=datahw2[which(datahw2$V17==3|datahw2$V17==8),]
head(pendigit38)
```

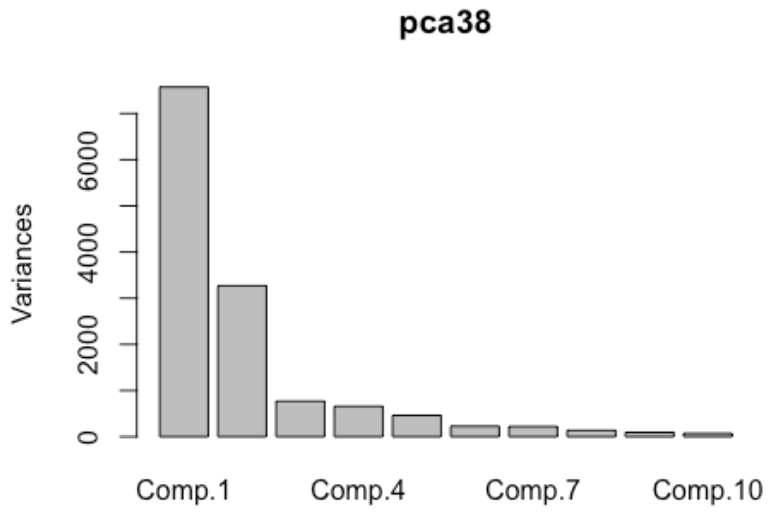
```
##    V1  V2 V3  V4  V5 V6  V7 V8  V9 V10 V11 V12 V13 V14 V15 V16 V17
## 1  88  92  2  99  16 66  94 37  70   0   0  24  42  65 100 100   8
## 2  80 100 18  98  60 66 100 29  42   0   0  23  42  61  56  98   8
## 3   0  94  9  57  20 19   7  0  20  36  70  68 100 100  18  92   8
## 17 30  74 55 100  89 87  66 56 100  38  92   8  41   0   0  20   3
## 22  0  76 30  48  53  9  11  0  47  34  97  66 100 100  38  85   8
## 32 41  84 73 100 100 82  62 60  97  38  91   8  42   0   0  19   3
```

```
pca38=princomp(pendigit38[,1:17])
summary(pca38)
```

```
## Importance of components:
##                             Comp.1     Comp.2     Comp.3      Comp.4
## Standard deviation     87.0271347 57.1915800 27.7253565 25.65147413
## Proportion of Variance  0.5557019  0.2399920  0.0564010  0.04827888
## Cumulative Proportion   0.5557019  0.7956939  0.8520949  0.90037374
##                             Comp.5      Comp.6      Comp.7     Comp.8
## Standard deviation     21.53583724 15.10003307 14.86655408 11.7279561
## Proportion of Variance  0.03402954  0.01672971  0.01621635  0.0100920
## Cumulative Proportion   0.93440327  0.95113298  0.96734933  0.9774413
##                             Comp.9     Comp.10     Comp.11     Comp.12
## Standard deviation     9.730838083 8.519071302 6.582821285 6.099003783
## Proportion of Variance 0.006947571 0.005324968 0.003179484 0.002729294
## Cumulative Proportion  0.984388901 0.989713869 0.992893353 0.995622647
##                            Comp.13     Comp.14      Comp.15      Comp.16
## Standard deviation     5.350491468 4.272723544 2.8334418486 2.1549179208
## Proportion of Variance 0.002100486 0.001339498 0.0005890621 0.0003407171
## Cumulative Proportion  0.997723133 0.999062631 0.9996516932 0.9999924103
```

```
##                             Comp.17
## Standard deviation     3.216216e-01
## Proportion of Variance 7.589670e-06
## Cumulative Proportion  1.000000e+00
```

```
par(mfrow=c(1,1))
screeplot(pca38)
```




```
 (pca38$sdev)^2
```

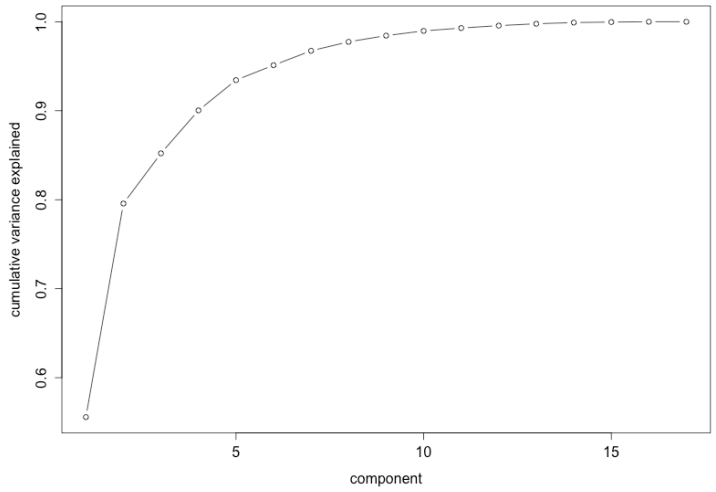
```
##      Comp.1     Comp.2       Comp.3       Comp.4       Comp.5     Comp.6
## 7573.7221744 3270.8768254 768.6953917 657.9981249 463.7922855  228.0109987
##      Comp.7     Comp.8       Comp.9      Comp.10      Comp.11    Comp.12
## 221.0144302  137.5449553  94.6892098  72.5745759   43.3335361   37.197847
##     Comp.13      Comp.14      Comp.15      Comp.16      Comp.17
##  28.6277590   18.2561665    8.0283927    4.6436712    0.1034405
```

```
cumsum((pca38$sdev)^2)/sum((pca38$sdev)^2)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 0.5557019 0.7956939 0.8520949 0.9003737 0.9344033 0.9511330 0.9673493
##    Comp.8    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14
## 0.9774413 0.9843889 0.9897139 0.9928934 0.9956226 0.9977231 0.9990626
##   Comp.15   Comp.16   Comp.17
## 0.9996517 0.9999924 1.0000000
```

```
plot(seq(1:17),cumsum((pca38$sdev)^2)/sum((pca38$sdev)^2),type="b",xlab="comp
onent",ylab="cumulative variance explained")
```

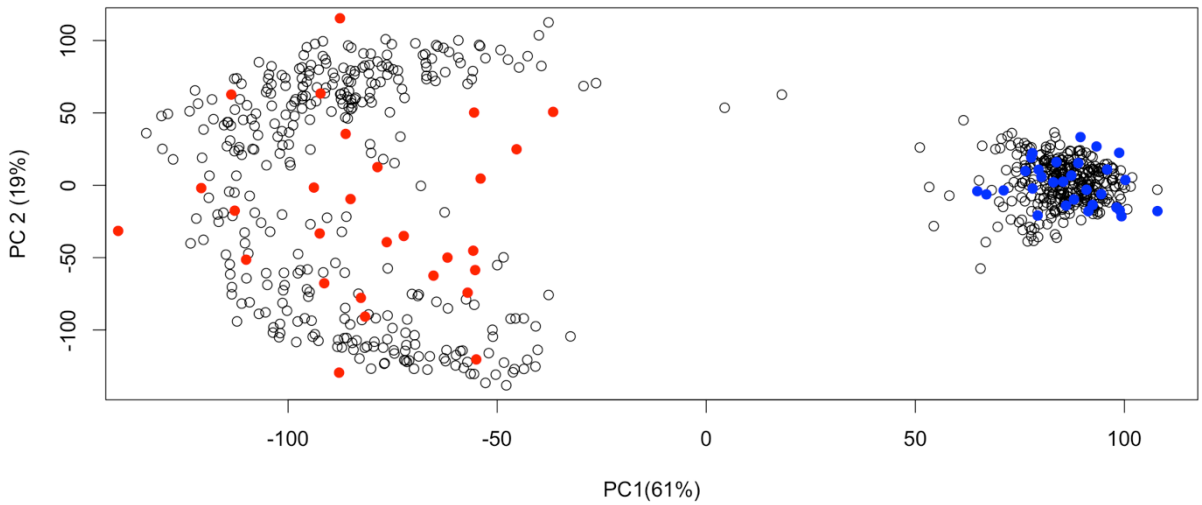
The cumulative variance of the components explained proportion reaches 90% at the first 4 components, which is faster than only performing PCA on class 3.



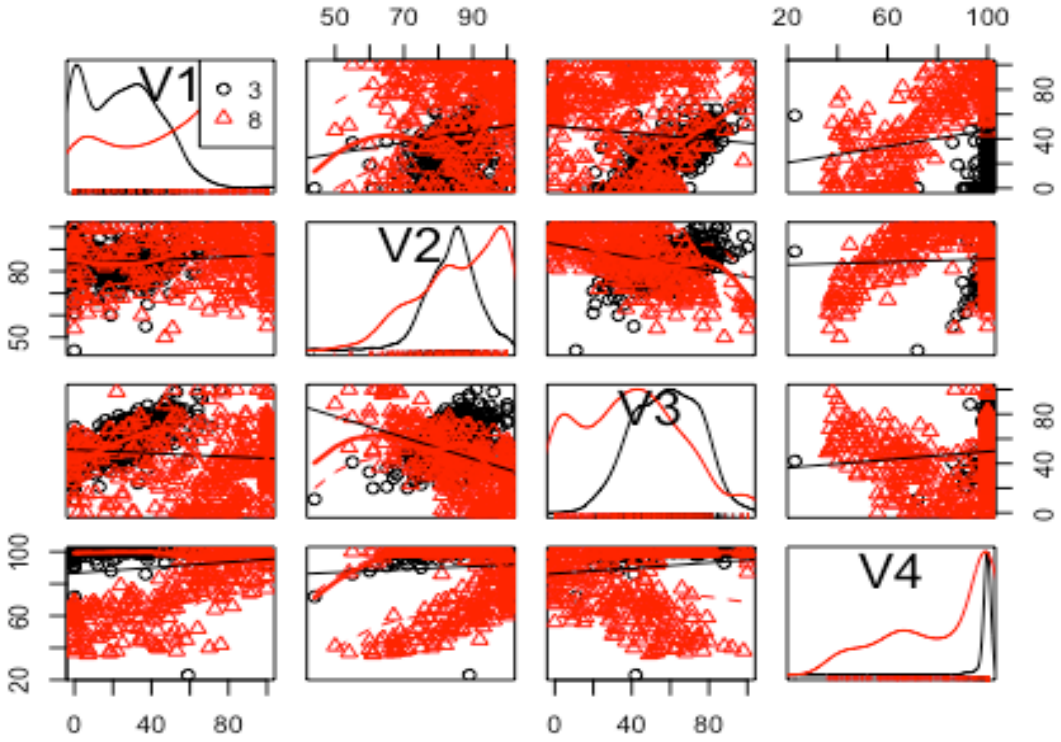

```
colmean8<-colMeans(pendigit38[which(pendigit38$V17==8),])
colmean3<-colMeans(pendigit38[which(pendigit38$V17==3),])
cov8<-cov(pendigit38[which(pendigit38$V17==8),])
cov3<-cov(pendigit38[which(pendigit38$V17==3),])
require(MASS)
set.seed(1)
n<-30
new8<-mvrnorm(n,colmean8,cov8)
new3<-mvrnorm(n,colmean3,cov3)
pred8<-predict(pca38,new8)
pred3<-predict(pca38,new3)

plot(pca38$x[,1],pca38$x[,2],cex=1,
xlab=paste0("PC",1,"(",round(pca38$sdev[1]/sum(pca$sdev)*100,0),"%)"),
ylab=paste0("PC ",2,"(", round(pca38$sdev[2]/sum(pca38$sdev)*100,0), "%)"))
points(pred8[,1],pred8[,2],col='red',pch=19)
points(pred3[,1],pred3[,2],col='blue',pch=19)
```

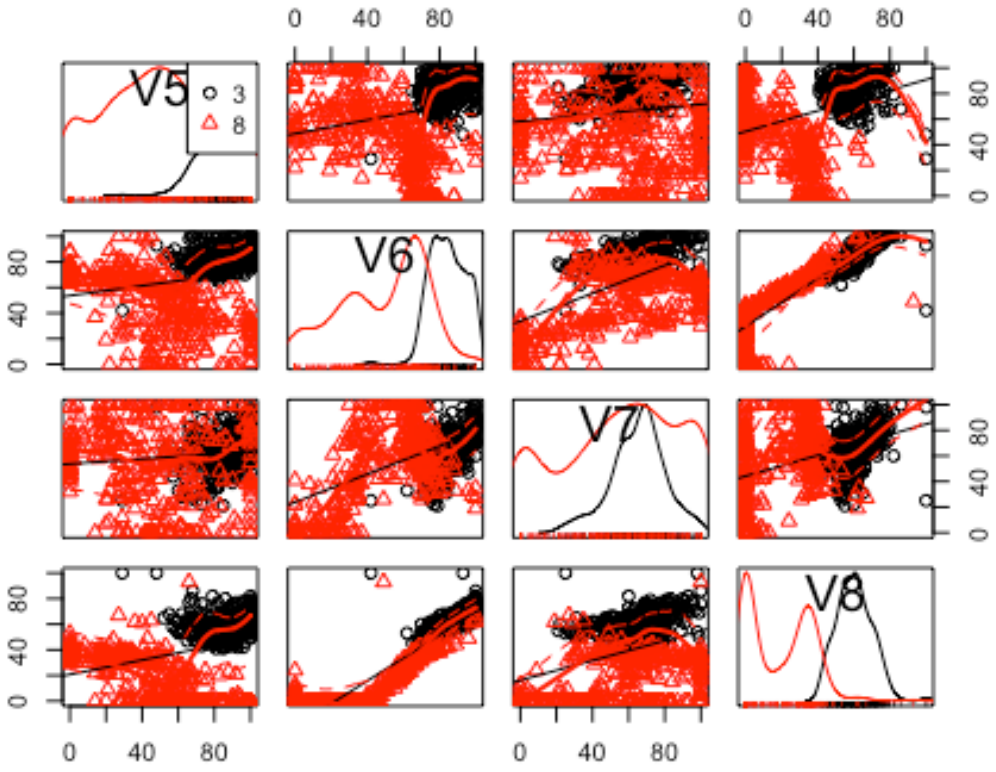
The hollow points in the plot below are the original data suggesting digit 3 or 8. And the solid points are new data created randomly. The red dots are from digit 8, and the blue dots are from digit3. As they are separated distinguishably, PCA is a good method to separate them.



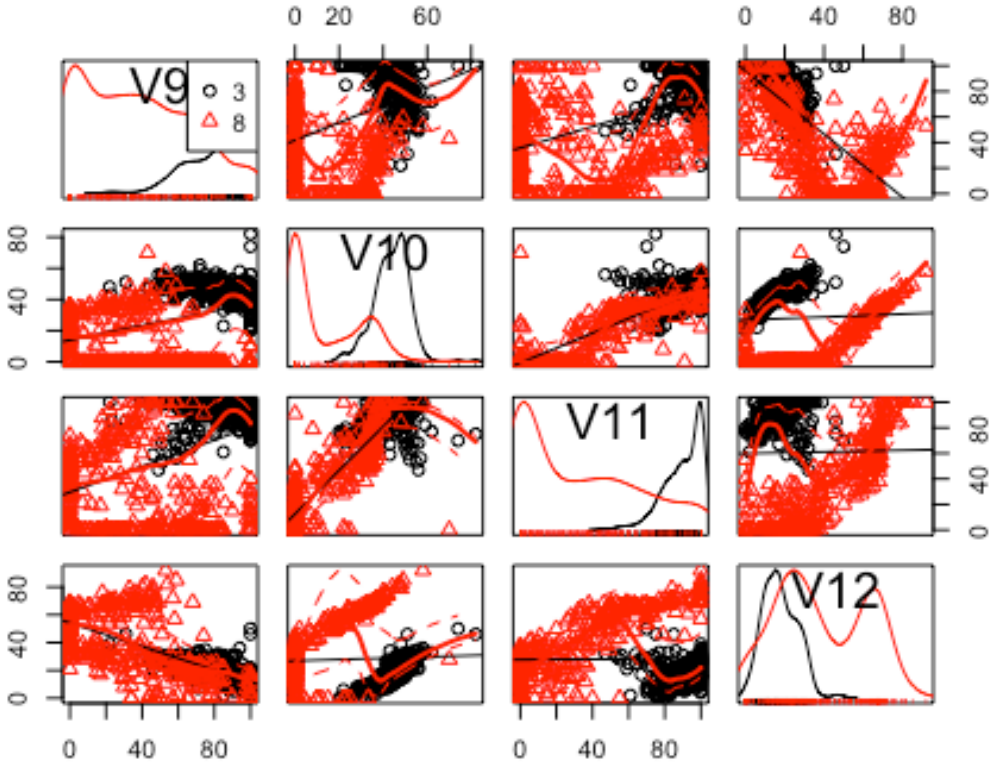
```
scatterplot.matrix(~V1+V2+V3+V4|V17,data=pendigit38)
```



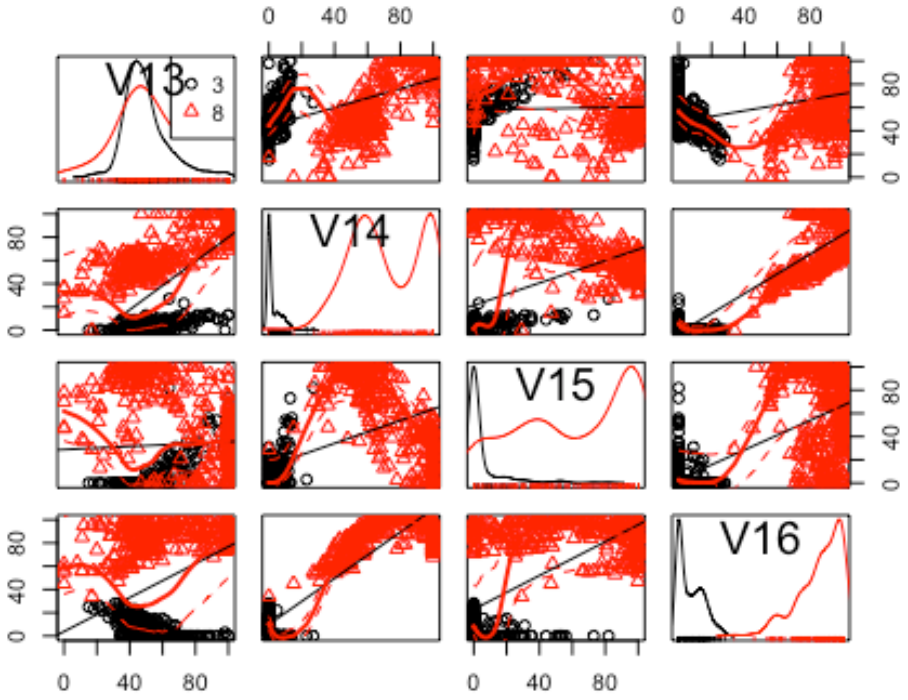

```
scatterplot.matrix(~V5+V6+V7+V8|V17,data=pendigit38)
```




```
scatterplot.matrix(~V9+V10+V11+V12|V17,data=pendigit38)
```

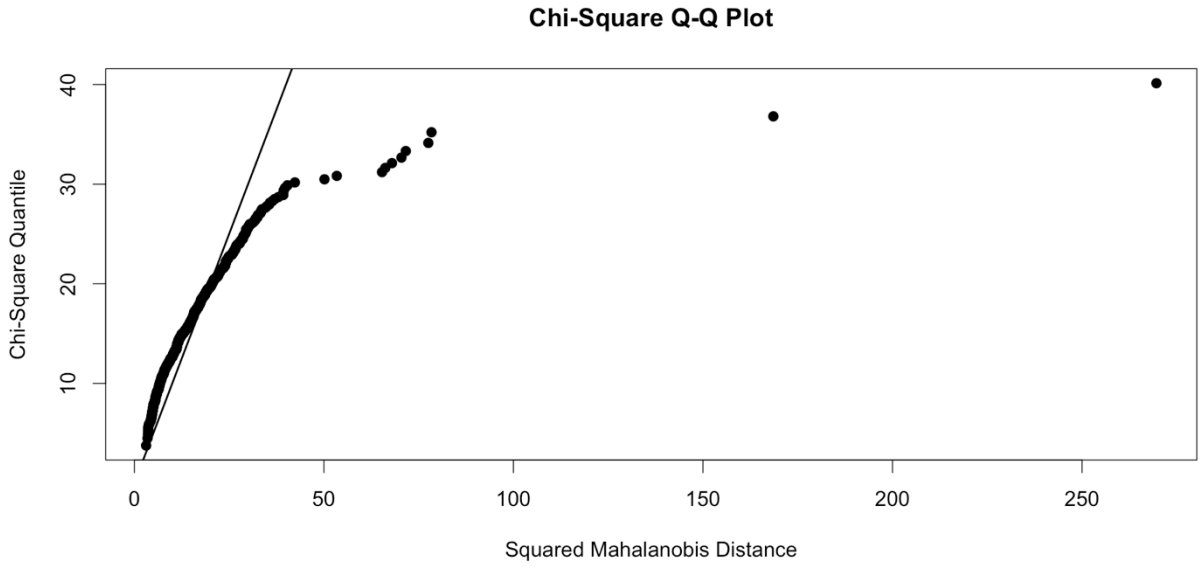



```
scatterplot.matrix(~V13+V14+V15+V16|V17,data=pendigit38)
```



From the plots shown above, I can see that for the V16 variable, the mass of points from digit 3 is closer to 0 and while those from digit 8 is closer to 100. And they have a very small overlapping area. So it is better to separate the class by looking at the V16 distribution. If more variables taken into account, I would suggest V14, V8, V11 as they have smaller overlapping area compared with others.

```
hzTest(pendigit38[,1:16],cov = TRUE,qqplot=TRUE)
```



From the Henze-Zirkler's Multivariate Normality Test, the data from digit 3 and 8 is not Multivariate Normal Distribution either.