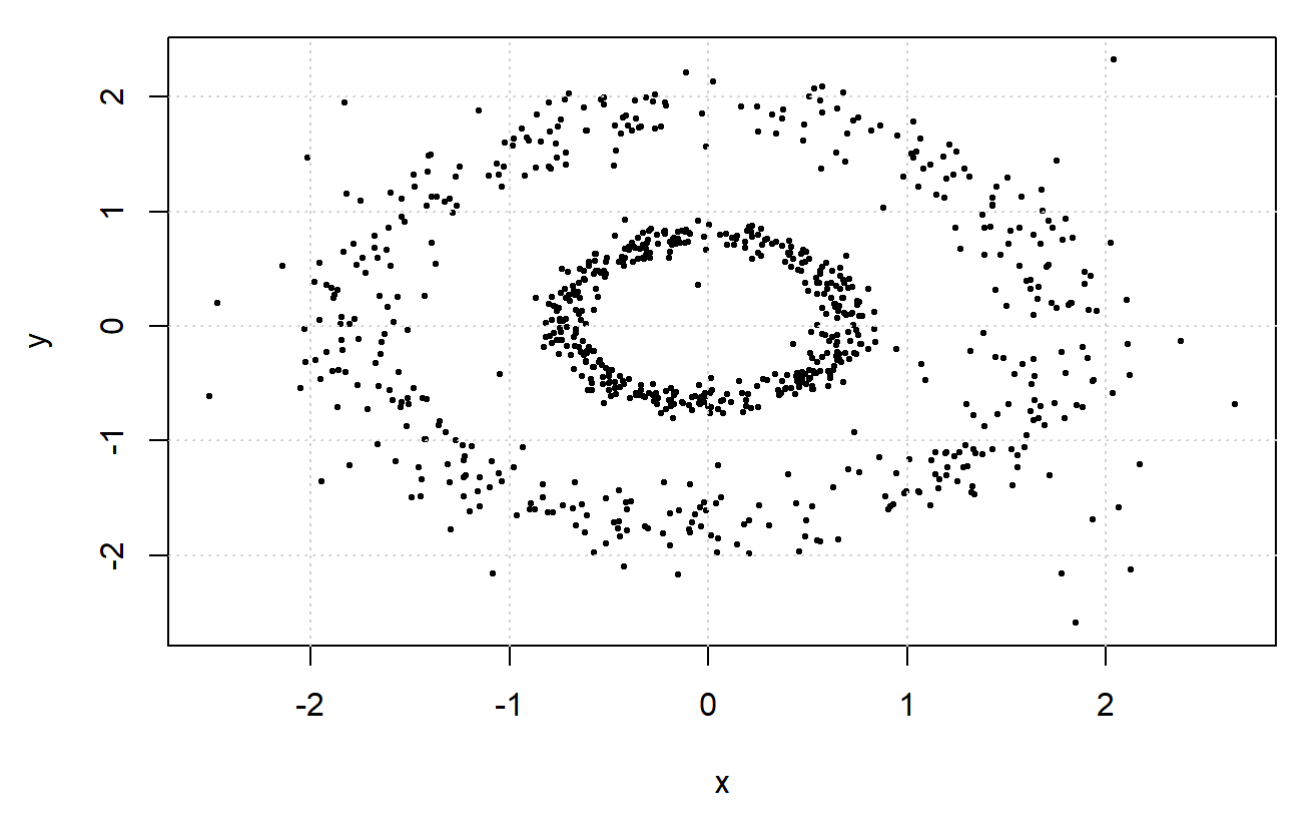
DBSCAN_G

circles dataset

```r
# DBSCAN.r
#
# install.packages("factoextra")
library(factoextra)
#
df = read.csv("circles.csv")
str(df)
```

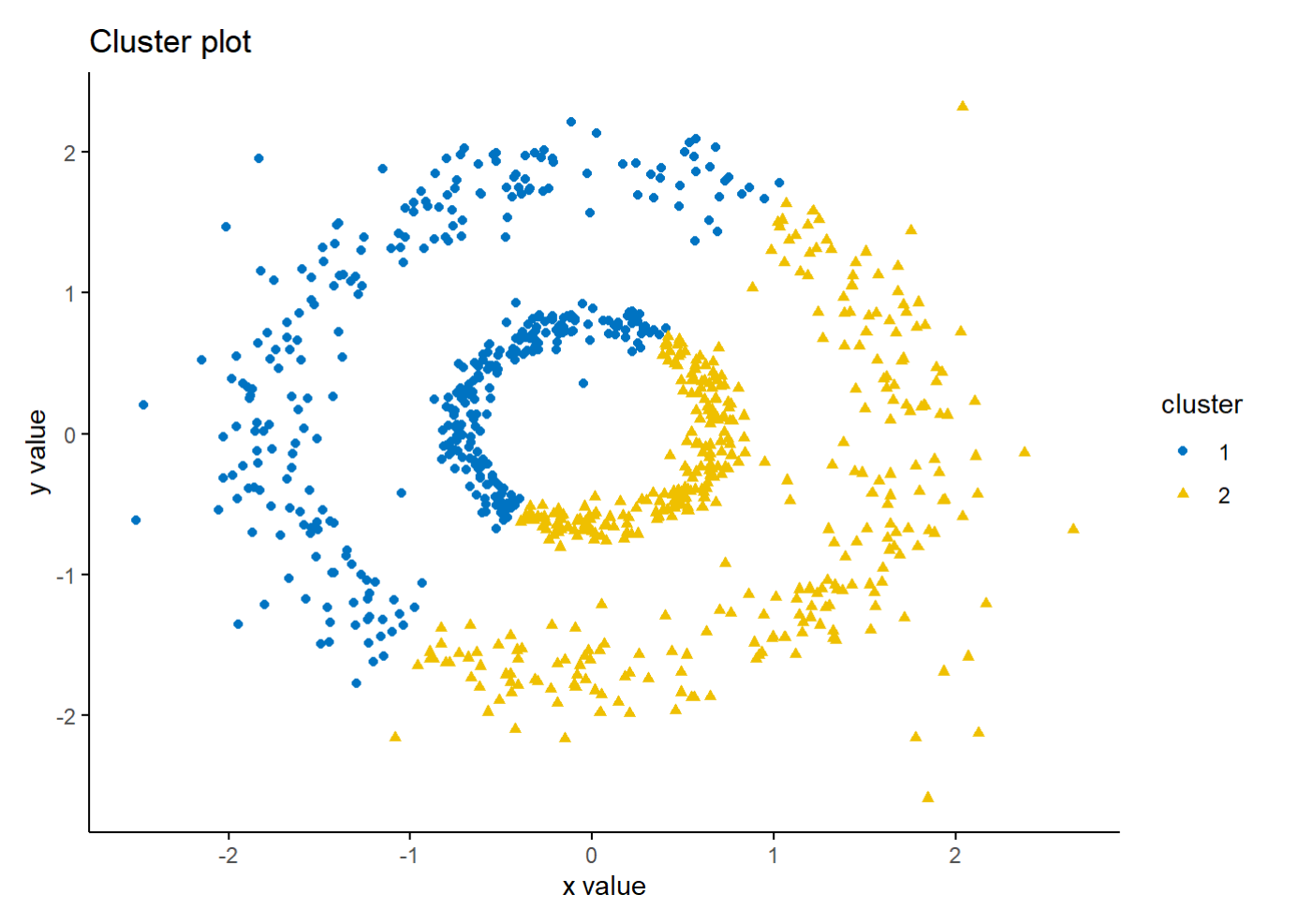
```
## 'data.frame':    833 obs. of  2 variables:
##  $ x: num  -0.804 0.853 0.927 -0.753 0.707 ...
##  $ y: num  -0.853 0.368 -0.275 -0.512 0.811 ...
```

```r
df0 = scale(df)
plot(y~x,df0,pch=19,cex=0.4)
grid()
```

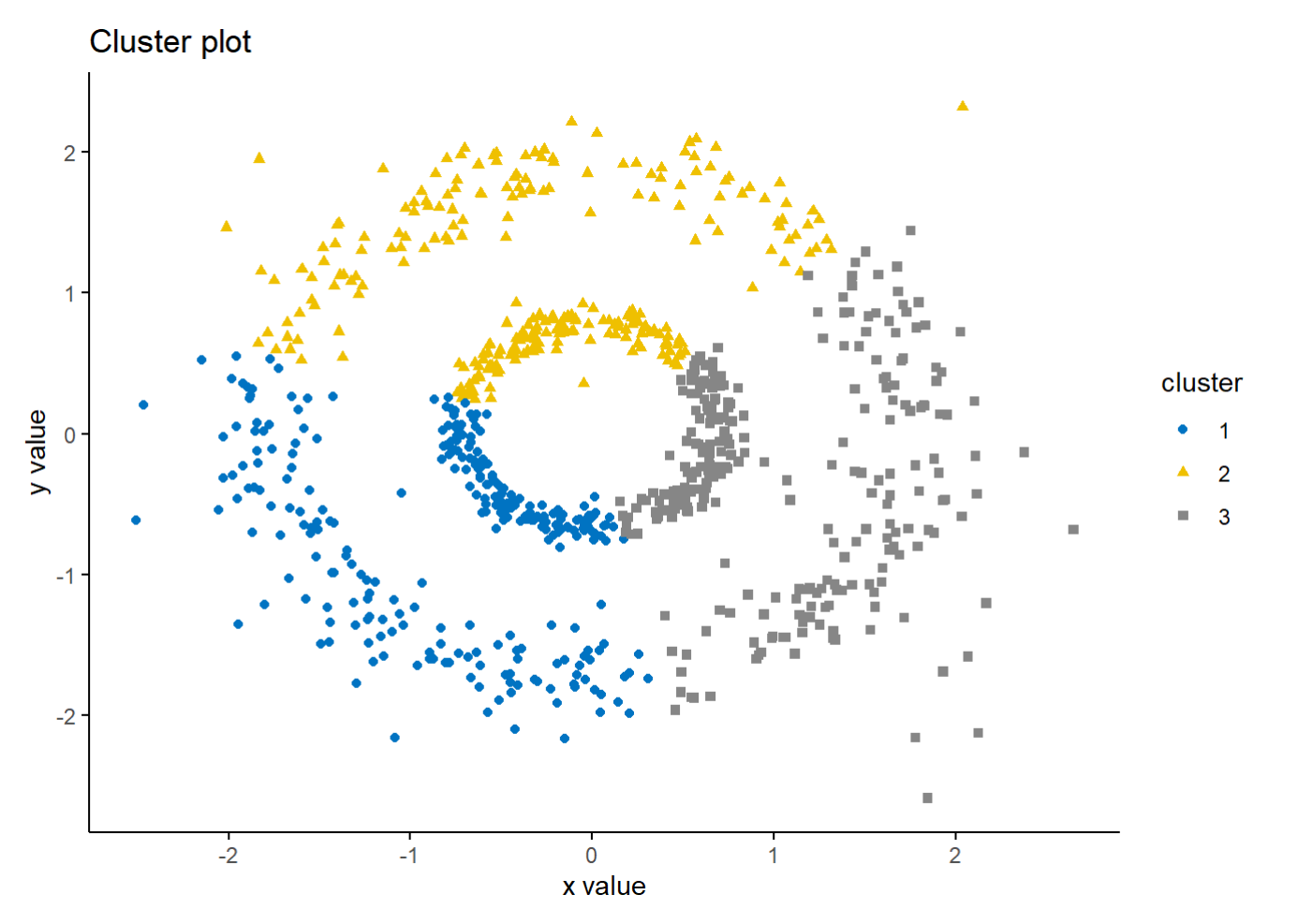



```r
#
```

```r
# kmeans with two clusters
#
set.seed(123)
kmodel2 = kmeans(df0, centers = 2, nstart = 25)
fviz_cluster(kmodel2, df0, geom = "point", show.clust.cent = F, ellipse= F,
             palette = "jco", ggtheme = theme_classic())
```




```r
#
# kmeans with three clusters
#
set.seed(123)
kmodel3 = kmeans(df0, centers = 3, nstart = 25)
fviz_cluster(kmodel3, df0, geom = "point", show.clust.cent = F, ellipse= F,
             palette = "jco", ggtheme = theme_classic())
```




```r
# Two libraries for DBSCAN

# install.packages("fpc")
library("fpc")   # dbscan()
#
# DBSCAN with eps = 0.18, MinPts = 3
#
set.seed(123)
db = dbscan(df0, eps = 0.18, MinPts = 5)
str(db)
```

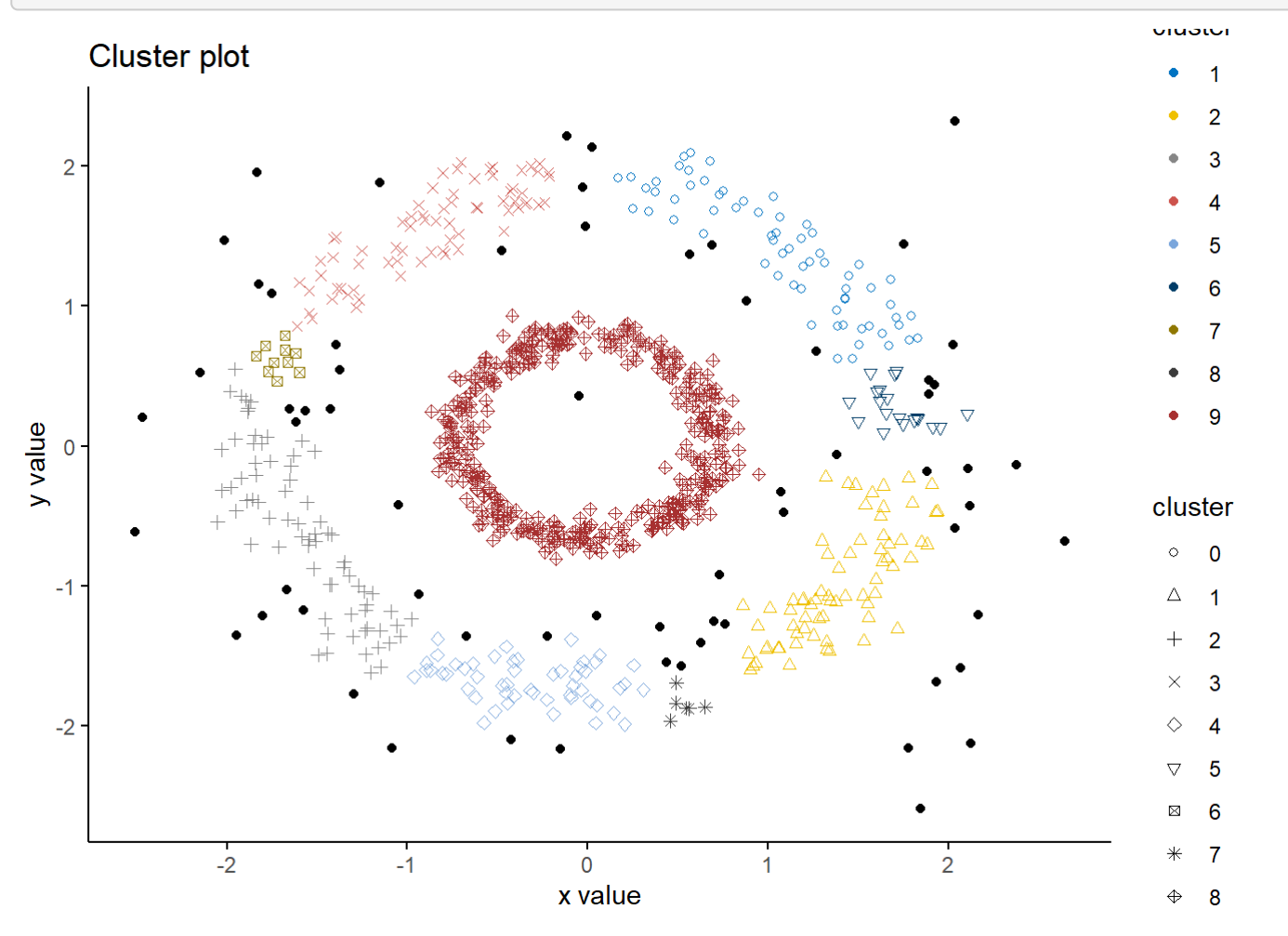
```
## List of 4
##  $ cluster: num [1:833] 3 1 2 3 1 1 4 5 3 3 ...
##  $ eps    : num 0.18
##  $ MinPts : num 5
##  $ isseed : logi [1:833] FALSE TRUE TRUE TRUE TRUE TRUE ...
##  - attr(*, "class")= chr "dbscan"
```

```r
print(db)
```

```
## dbscan Pts=833 MinPts=5 eps=0.18
##         0  1  2  3  4  5  6  7 8   9
## border 65 12 14 13  8  8  4  0 3   1
## seed    0 53 55 60 62 44 15 10 3 403
## total  65 65 69 73 70 52 19 10 6 404
```

```r
# clusters points are arranged by columns
# column 0 includes noise points
# seed are core points
#

fviz_cluster(db, data = df0, geom = "point", show.clust.cent = F,
             ellipse = F, stand = F, palette = "jco",
             ggtheme = theme_classic())
```




```r
# outliers are shown as black pts
#
```

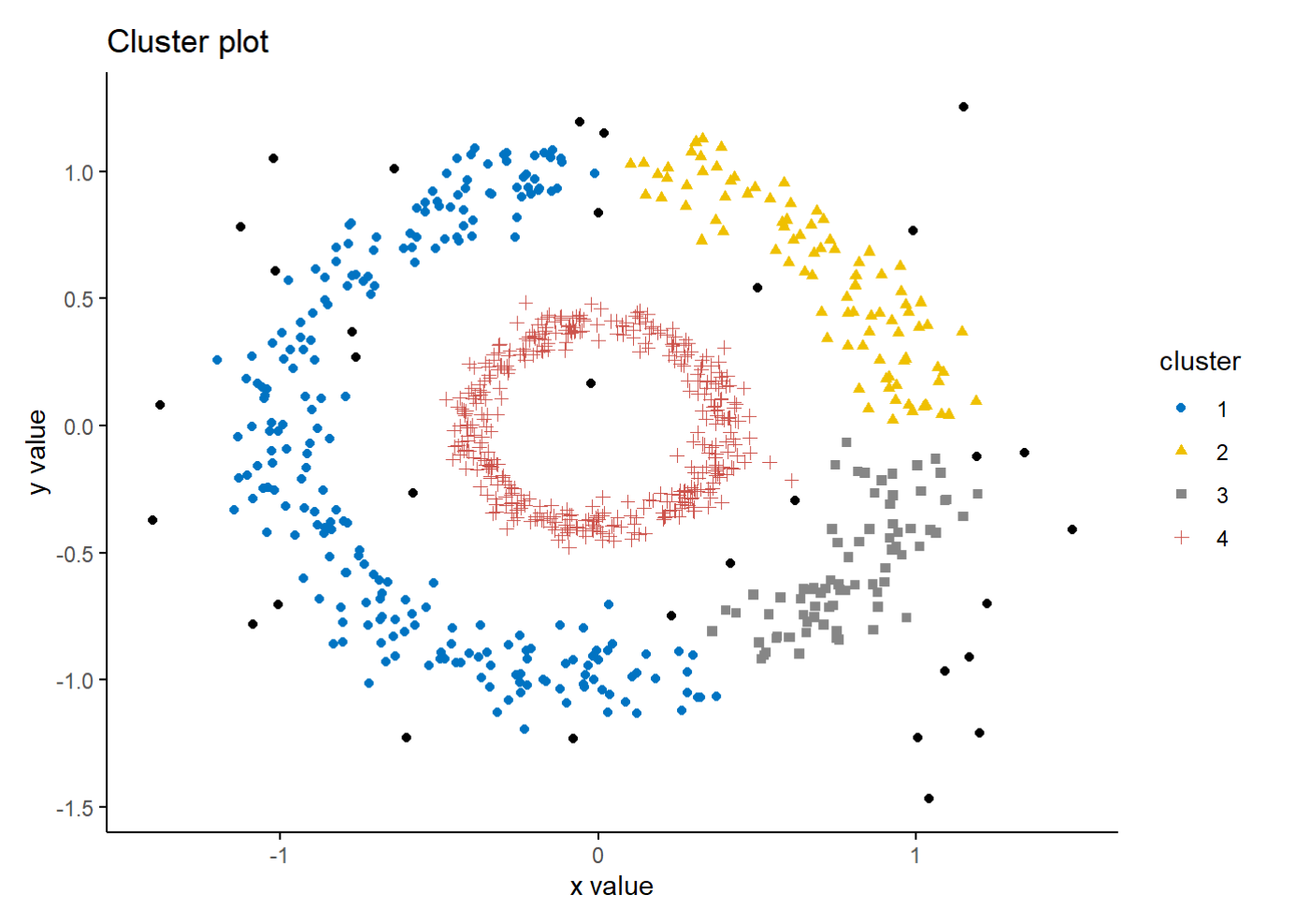
```r
# see cluster membership of some rows
db$cluster[sample(1:833, 20)]
```

```
##  [1] 9 9 0 9 6 0 1 1 4 2 5 4 9 9 9 9 9 4 3 9
```

```r
#
# how sensible is DBSCAN to eps?
#
# DBSCAN with eps = 0.22, with same MinPts = 5
#
set.seed(123)
db = dbscan(df0, eps = 0.22, MinPts = 5)
print(db)
```

```
## dbscan Pts=833 MinPts=5 eps=0.22
##         0   1  2  3   4
## border 32  20  9  7   1
## seed    0 209 82 69 404
## total  32 229 91 76 405
```

```r
fviz_cluster(db, data = df, geom = "point", show.clust.cent = F,
             ellipse = F, stand = F, palette = "jco",
             ggtheme = theme_classic())
```

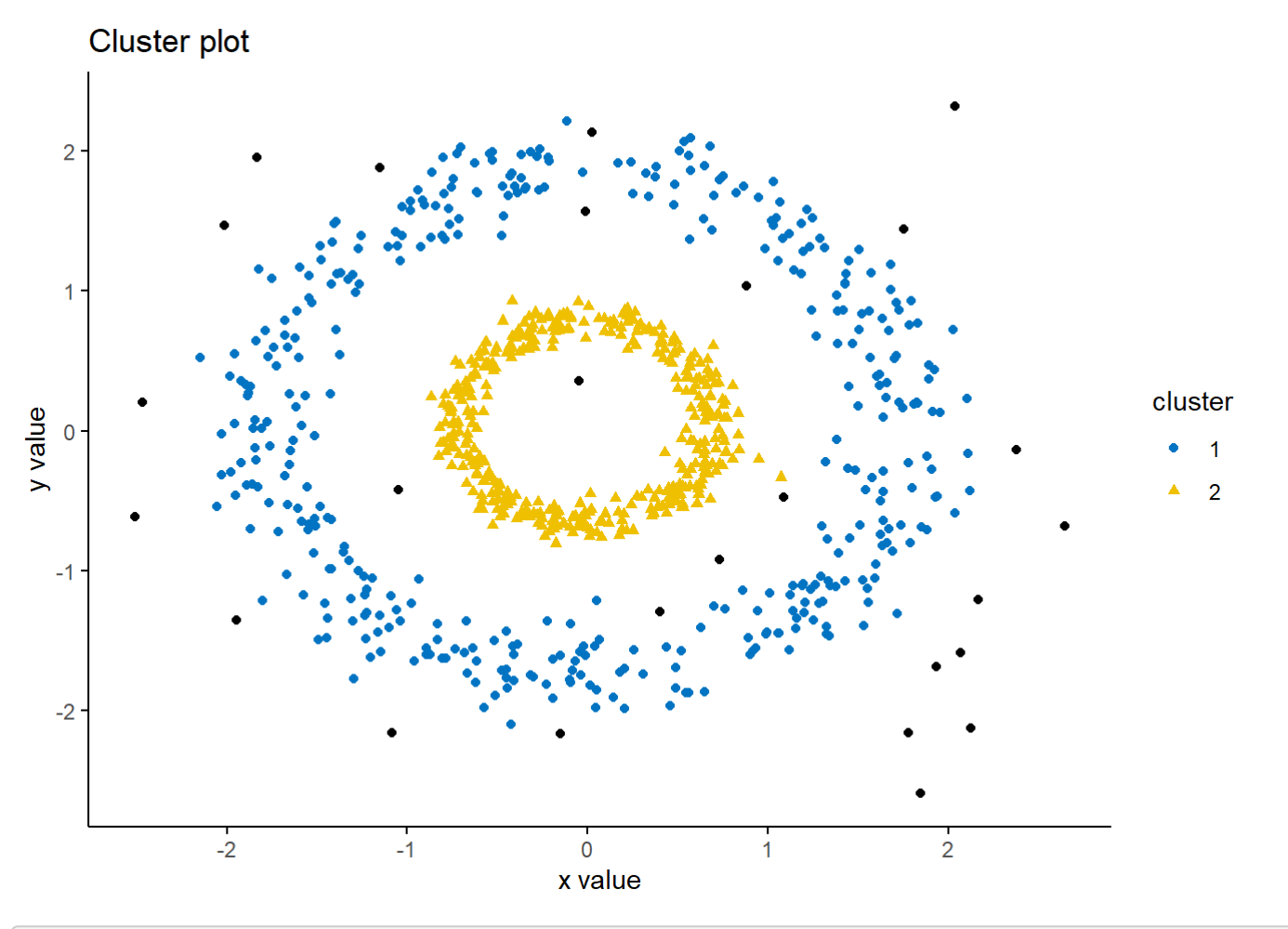



```r
#
```

```r
set.seed(123)
db = dbscan(df0, eps = 0.25, MinPts = 5)
print(db)
```

```
## dbscan Pts=833 MinPts=5 eps=0.25
##         0   1   2
## border 26  26   1
## seed    0 376 404
## total  26 402 405
```

```r
fviz_cluster(db, data = df0, geom = "point", show.clust.cent = F,
             ellipse = F, stand = F, palette = "jco",
             ggtheme = theme_classic())
```




```r
# clusters 1,2,3 should be a single cluster
#
# Find best epsilon given k = MinPts
#
# install.packages("dbscan")
```

```r
library("dbscan")     # kNNdistplot()
```

```
##
## 载入程辑包：'dbscan'
```

```
## The following object is masked from 'package:fpc':
##
##     dbscan
```

```
## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

```r
#
# distance to the 5-th nearest neighbor
vector1 = kNNdist(df0,k = 5)
length(vector1)
```

```
## [1] 833
```

```r
head(vector1)
```

```
## [1] 0.2476032 0.1562298 0.1612715 0.1556517 0.2017160 0.1905476
```

```r
# distances to the 1st, 2nd, ..., k-th nearest neighbors
matrix2 = kNNdist(df0,k = 5,all=T)
head(matrix2)
```

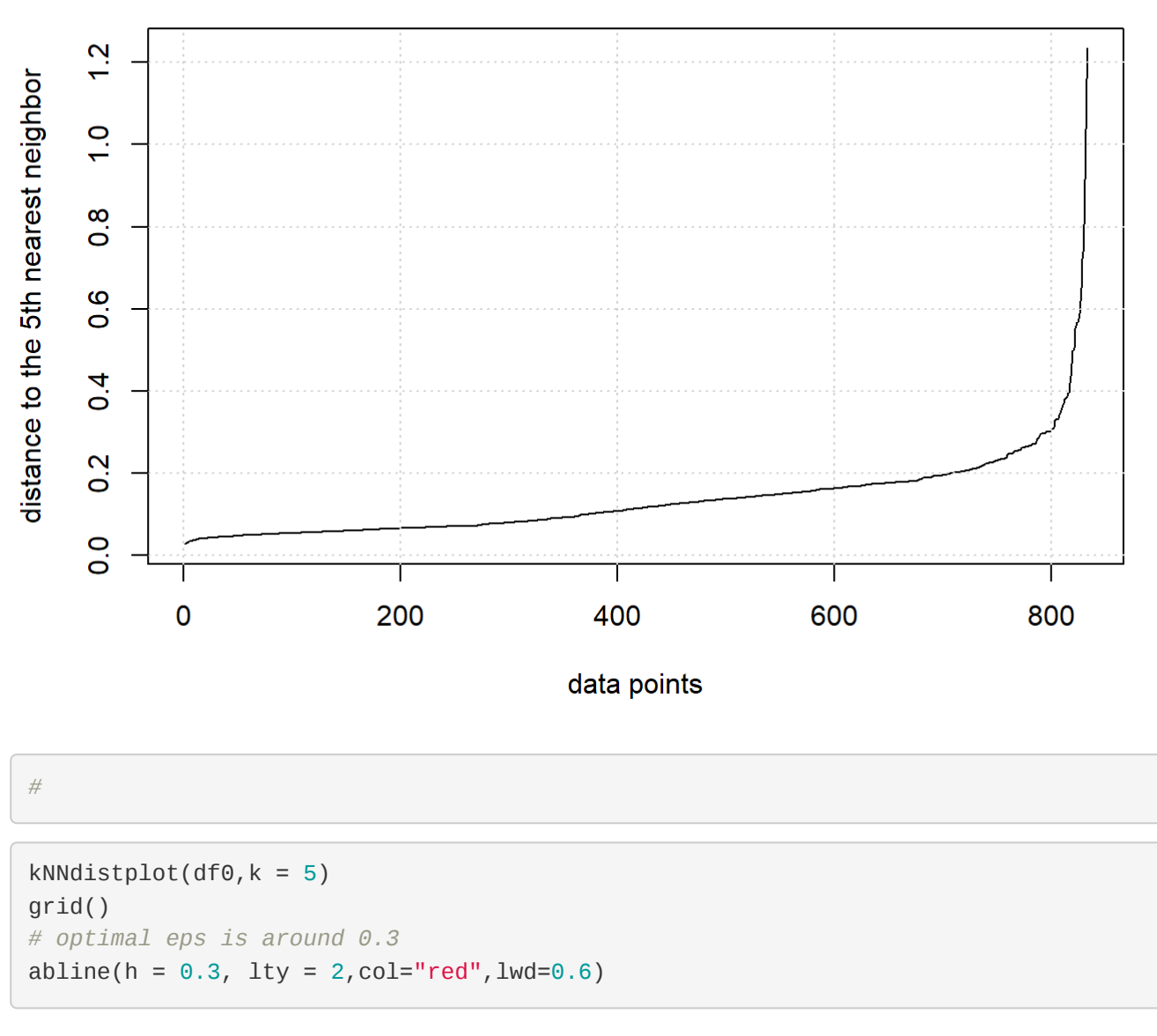
```
##               1          2         3         4         5
## [1,] 0.04982494 0.14247034 0.1887508 0.2155387 0.2476032
## [2,] 0.10847365 0.11207032 0.1462333 0.1479911 0.1562298
## [3,] 0.06582546 0.10309930 0.1222139 0.1529488 0.1612715
## [4,] 0.03710787 0.06927237 0.1414263 0.1447587 0.1556517
## [5,] 0.06887157 0.07547471 0.1546196 0.1735182 0.2017160
## [6,] 0.05189067 0.13764014 0.1652928 0.1689794 0.1905476
```

```r
dim(matrix2)
```

```
## [1] 833   5
```

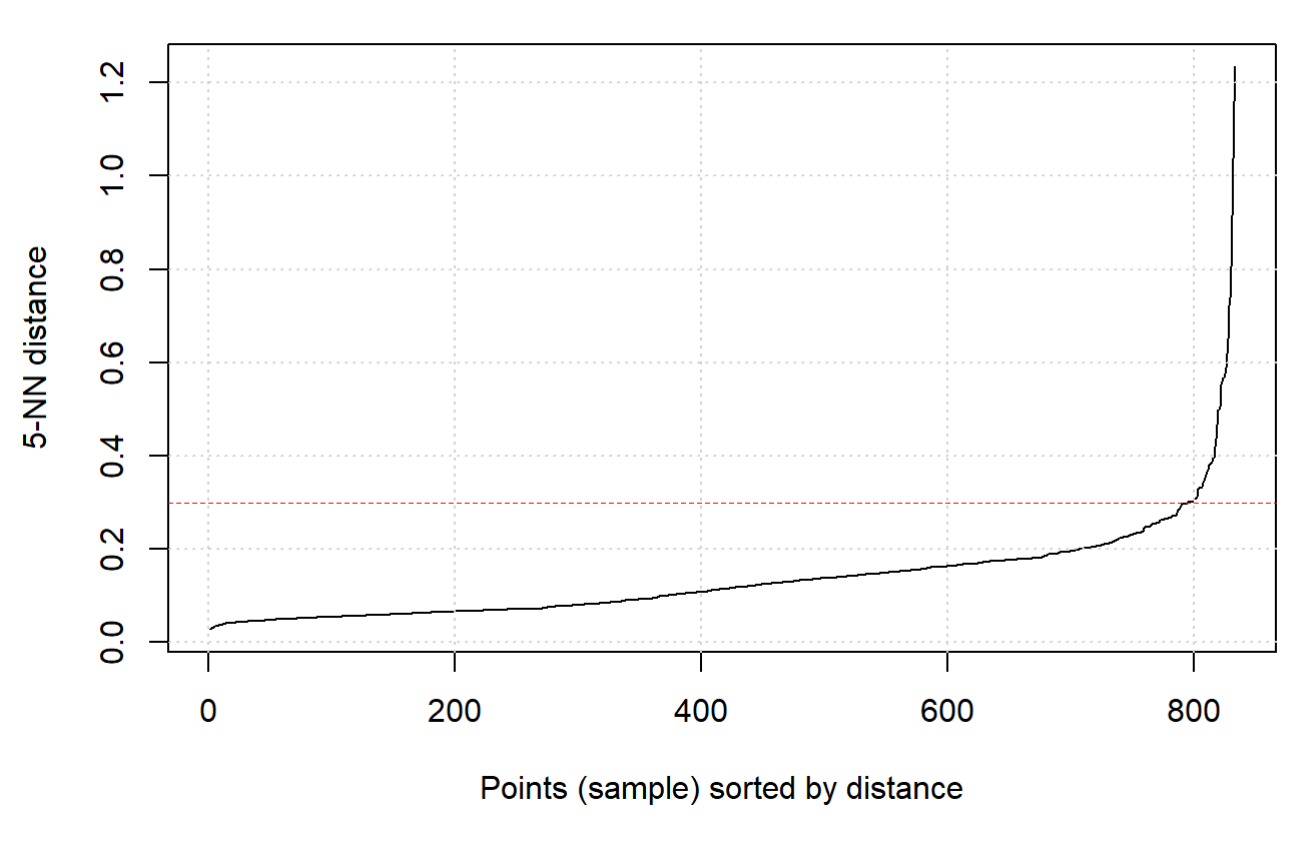
```r
# vector1 is the last column in matrix2
#
```

```r
# plot distances to the 5th nearest neighbor (sorted)
plot(sort(vector1),type="l",xlab="data points",
     ylab="distance to the 5th nearest neighbor")
grid()
```




```r
#
```

```r
kNNdistplot(df0,k = 5)
grid()
# optimal eps is around 0.3
abline(h = 0.3, lty = 2,col="red",lwd=0.6)
```




```r
#
```