Homework 5

Jingyi Guo (jingyig1), Pittsburgh Campus

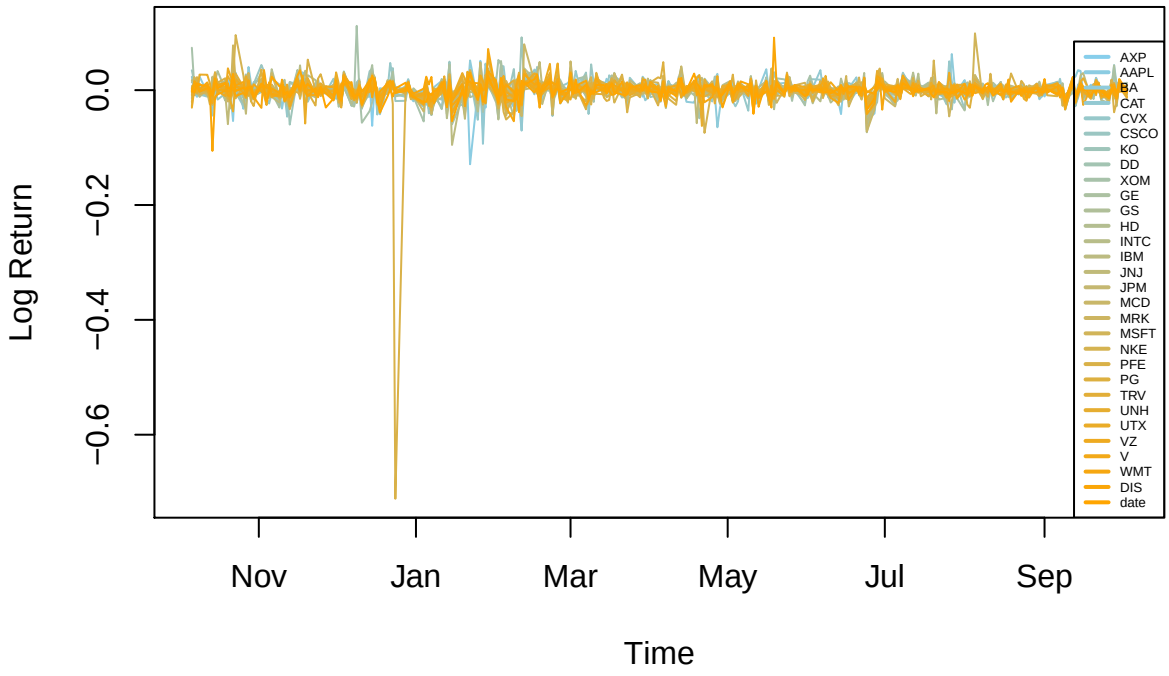
12/5/2017

```
setwd("~/Desktop/ML2/Homework 5")
load('djia_data.RData')
load('djia_info.RData')
source('djia_helpers.R')
library("protoclust")
```

1

(a)

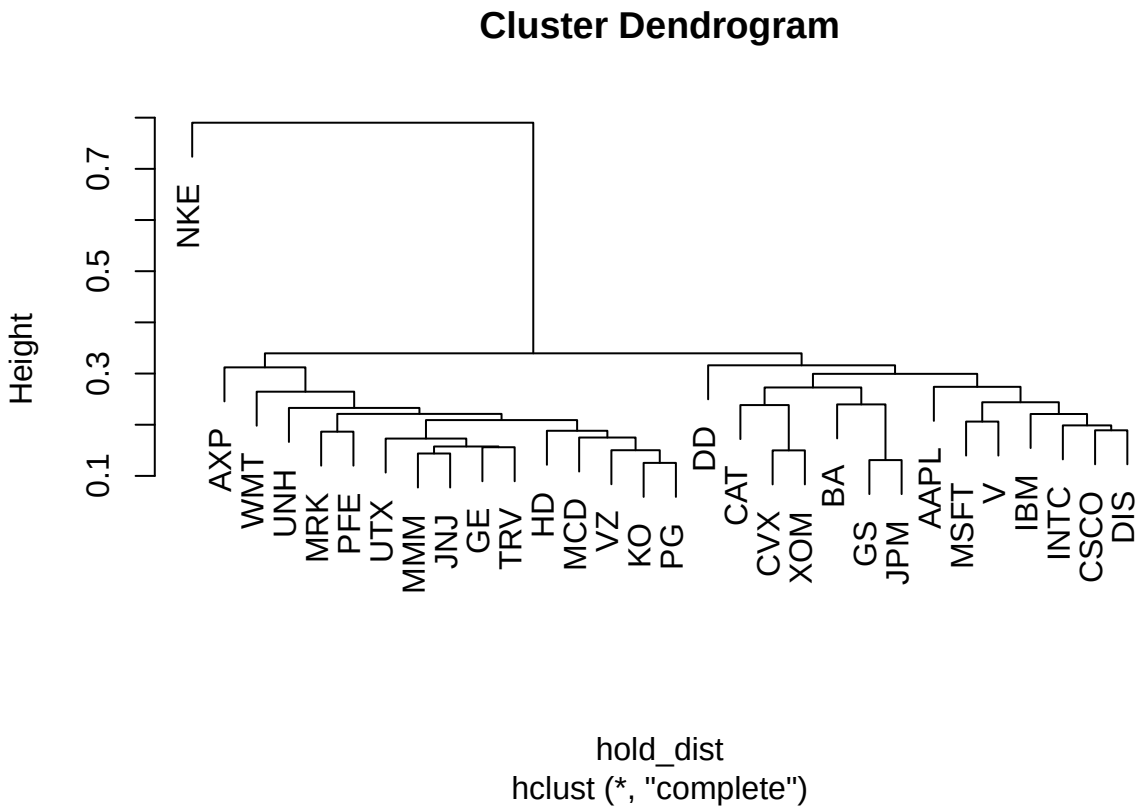
```
palette = colorRampPalette(c('skyblue','orange'))(30)
djia_close_log = log(djia_close[-1,1:30]/djia_close[-nrow(djia_close),1:30])
plot(djia_close$date[-1], djia_close_log[,1], type="l", ylim=c(min(djia_close_log), max(djia_close_log))
for (i in 2:30) {
  lines(djia_close$date[-1], djia_close_log[,i], col=palette[i])
}
legend("bottomright", legend=colnames(djia_close)[-1], lwd=2, col=palette, cex=0.4)
```



(b)

```
hold_dist = dist(t(as.matrix(djia_close_log)))
hold_clust1 = hclust(hold_dist, method="complete")
```

```r
plot(hold_clust1)
```



According to the dendrogram above, grouping the stocks into 3 clusters is reasonable, where the majority of stocks are divided into 2 large subgroups with height=0.3, and NKE being very different from other clusters.

(c)

```r
k=3
hold_cut1 = cutree(hold_clust1, k=k)
hold_group1 = resolve_groups(hold_cut1, djia_info)
for (i in 1:k) {
  print(paste("Group", i))
  print(hold_group1$industry[[i]])
  print(hold_group1$symbol[[i]])
}
```

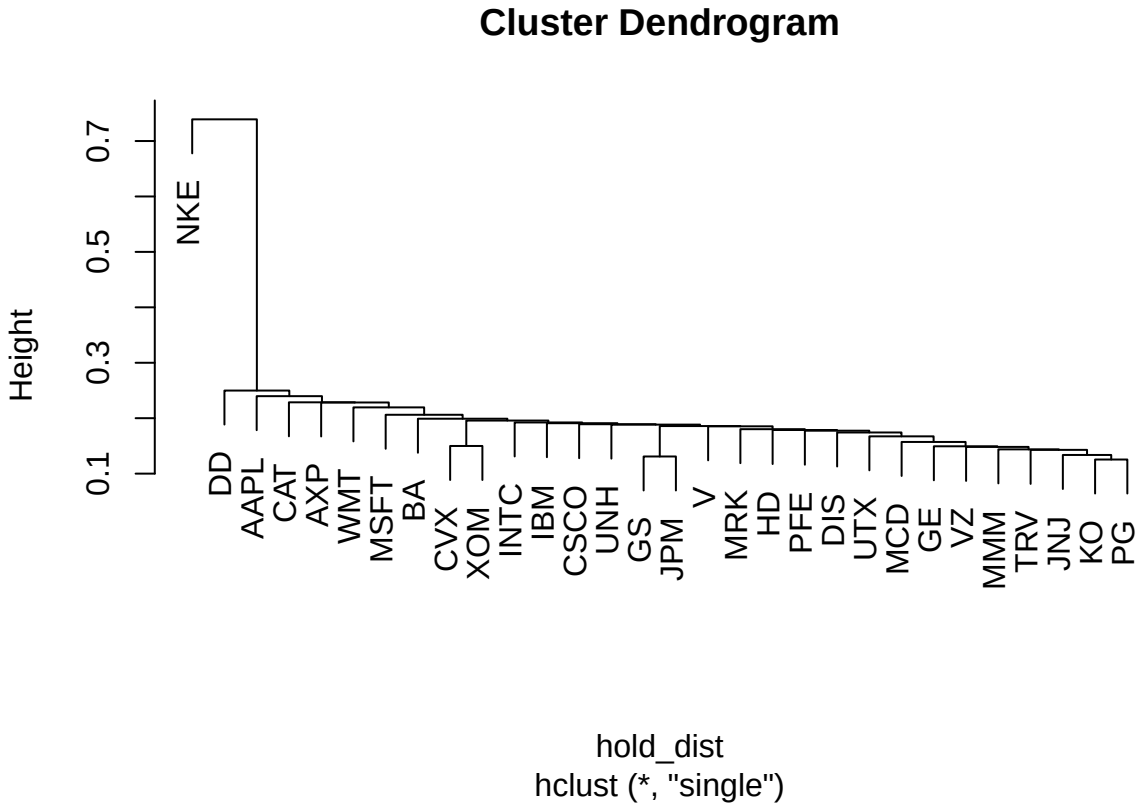
```
## [1] "Group 1"
##  [1] "Conglomerate"              "Consumer finance"
##  [3] "Beverages"                 "Conglomerate"
##  [5] "Home improvement retailer" "Pharmaceuticals"
##  [7] "Fast food"                 "Pharmaceuticals"
##  [9] "Pharmaceuticals"           "Consumer goods"
## [11] "Insurance"                 "Managed health care"
## [13] "Conglomerate"              "Telecommunication"
## [15] "Retail"
##  [1] "MMM" "AXP" "KO"  "GE"  "HD"  "JNJ" "MCD" "MRK" "PFE" "PG"  "TRV"
## [12] "UNH" "UTX" "VZ"  "WMT"
```

```
## [1] "Group 2"
##  [1] "Consumer electronics"
##  [2] "Aerospace and defense"
##  [3] "Construction and mining equipment"
##  [4] "Oil & gas"
##  [5] "Computer networking"
##  [6] "Chemical industry"
##  [7] "Oil & gas"
##  [8] "Banking"
##  [9] "Semiconductors"
## [10] "Computers and technology"
## [11] "Banking"
## [12] "Consumer electronics"
## [13] "Consumer banking"
## [14] "Broadcasting and entertainment"
##  [1] "AAPL" "BA"   "CAT"  "CVX"  "CSCO" "DD"   "XOM"  "GS"   "INTC" "IBM"
## [11] "JPM"  "MSFT" "V"    "DIS"
## [1] "Group 3"
## [1] "Apparel"
## [1] "NKE"
```

The stocks in group 1 are more related to service industry and consumer's daily needs such as Fast Food and Pharmaceuticals. The stocks in group 2 are more related to manufactoring industry, such as Oil & gas and Aerospace and defense. Group 3 only consists of the stock NKE.

(d)

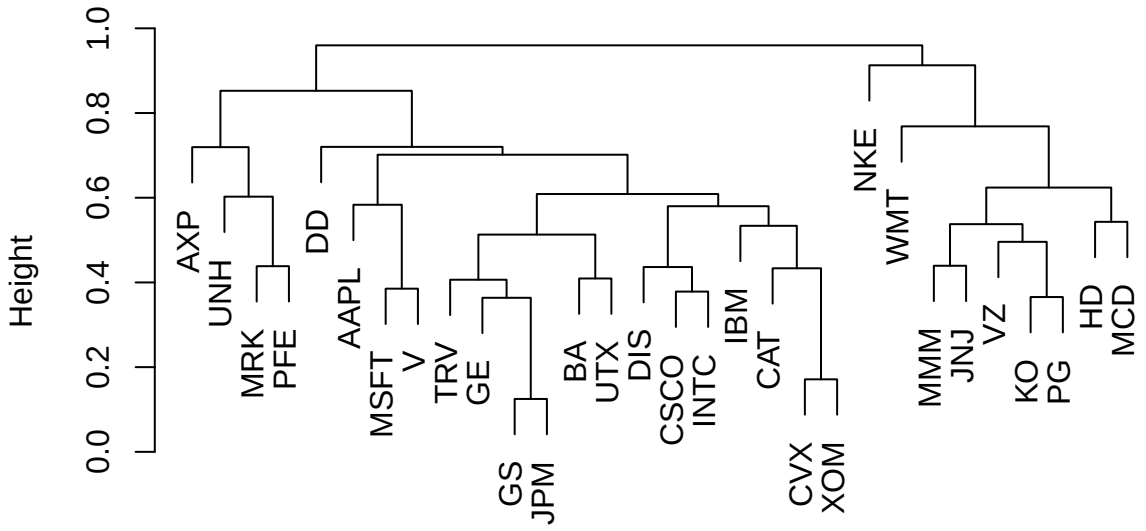
```
hold_clust2 = hclust(hold_dist, method="single")
plot(hold_clust2)
```



The clustering look worse than in (b) because there's no clear division of subgroups in the dendrogram.

(e)

```
k=4
hold_dist_corr = as.dist(1-cor(djia_close_log))
hold_clust3 = hclust(hold_dist_corr, method="complete")
plot(hold_clust3)
```



hold_dist_corr
hclust (*, "complete")

```
hold_cut3 = cutree(hold_clust3, k=k)
hold_group3 = resolve_groups(hold_cut3, djia_info)
for (i in 1:k) {
  print(paste("Group", i))
  print(hold_group3$industry[[i]])
  print(hold_group3$symbol[[i]])
}
```

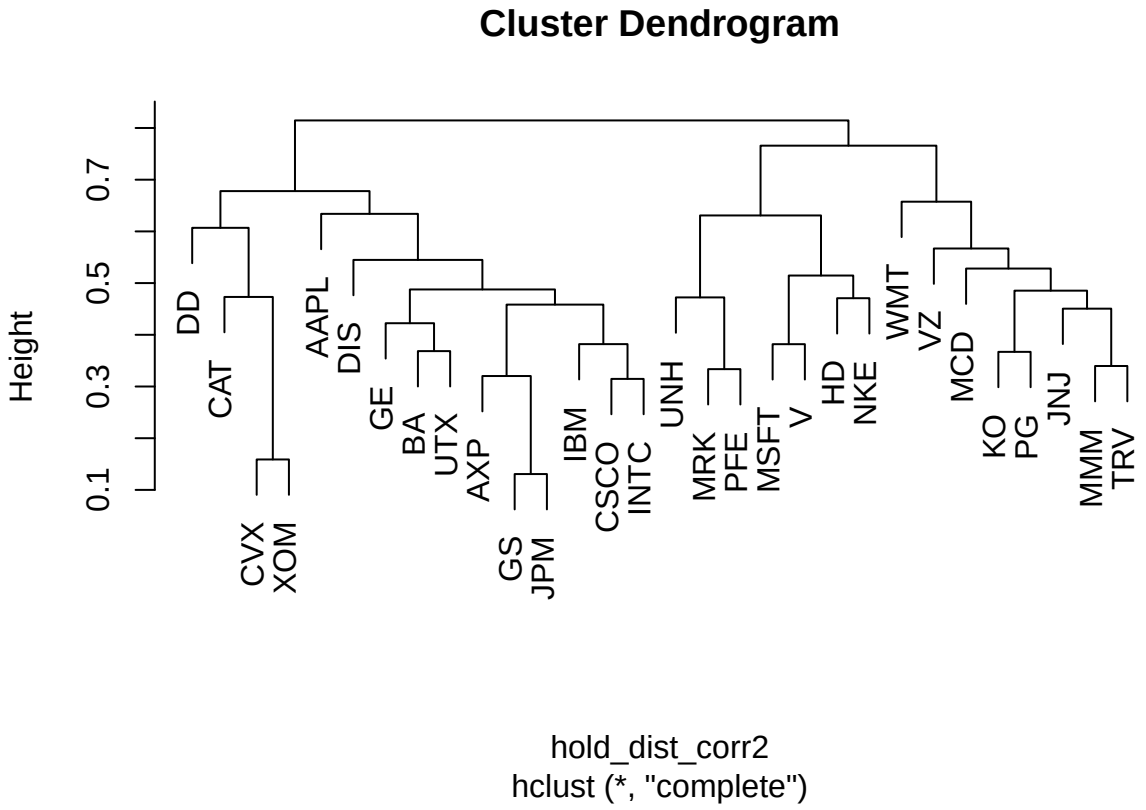
```
## [1] "Group 1"
## [1] "Conglomerate"              "Beverages"
## [3] "Home improvement retailer" "Pharmaceuticals"
## [5] "Fast food"                 "Consumer goods"
## [7] "Telecommunication"         "Retail"
## [1] "MMM" "KO"  "HD"  "JNJ" "MCD" "PG"  "VZ"  "WMT"
## [1] "Group 2"
## [1] "Consumer finance"    "Pharmaceuticals"     "Pharmaceuticals"
## [4] "Managed health care"
## [1] "AXP" "MRK" "PFE" "UNH"
## [1] "Group 3"
##  [1] "Consumer electronics"
##  [2] "Aerospace and defense"
##  [3] "Construction and mining equipment"
##  [4] "Oil & gas"
##  [5] "Computer networking"
##  [6] "Chemical industry"
##  [7] "Oil & gas"
##  [8] "Conglomerate"
##  [9] "Banking"
```

```
## [10] "Semiconductors"
## [11] "Computers and technology"
## [12] "Banking"
## [13] "Consumer electronics"
## [14] "Insurance"
## [15] "Conglomerate"
## [16] "Consumer banking"
## [17] "Broadcasting and entertainment"
##  [1] "AAPL" "BA"   "CAT"  "CVX"  "CSCO" "DD"   "XOM"  "GE"   "GS"   "INTC"
## [11] "IBM"  "JPM"  "MSFT" "TRV"  "UTX"  "V"    "DIS"
## [1] "Group 4"
## [1] "Apparel"
## [1] "NKE"
```

This result looks better. The 3 groups are well separated at height=0.7, although NKE is still in a single group. In this division, group 1 and group 2 are more related to the service industry while group 3 is more related to manufacturing industry.

(f)

```
hold_dist_corr2 = as.dist(1-cor(djia_close_log, method='spearman'))
hold_clust4 = hclust(hold_dist_corr2, method="complete")
plot(hold_clust4)
```




```
hold_cut4 = cutree(hold_clust4, k=k)
hold_group4 = resolve_groups(hold_cut4, djia_info)
for (i in 1:k) {
  print(paste("Group", i))
```

```
  print(hold_group4$industry[[i]])
  print(hold_group4$symbol[[i]])
}
```

```
## [1] "Group 1"
## [1] "Conglomerate"      "Beverages"         "Pharmaceuticals"
## [4] "Fast food"         "Consumer goods"    "Insurance"
## [7] "Telecommunication" "Retail"
## [1] "MMM" "KO"  "JNJ" "MCD" "PG"  "TRV" "VZ"  "WMT"
## [1] "Group 2"
##  [1] "Consumer finance"               "Consumer electronics"
##  [3] "Aerospace and defense"          "Computer networking"
##  [5] "Conglomerate"                   "Banking"
##  [7] "Semiconductors"                 "Computers and technology"
##  [9] "Banking"                        "Conglomerate"
## [11] "Broadcasting and entertainment"
##  [1] "AXP"  "AAPL" "BA"   "CSCO" "GE"   "GS"   "INTC" "IBM"  "JPM"  "UTX"
## [11] "DIS"
## [1] "Group 3"
## [1] "Construction and mining equipment" "Oil & gas"
## [3] "Chemical industry"                 "Oil & gas"
## [1] "CAT" "CVX" "DD"  "XOM"
## [1] "Group 4"
## [1] "Home improvement retailer" "Pharmaceuticals"
## [3] "Consumer electronics"      "Apparel"
## [5] "Pharmaceuticals"           "Managed health care"
## [7] "Consumer banking"
## [1] "HD"   "MRK"  "MSFT" "NKE"  "PFE"  "UNH"  "V"
```

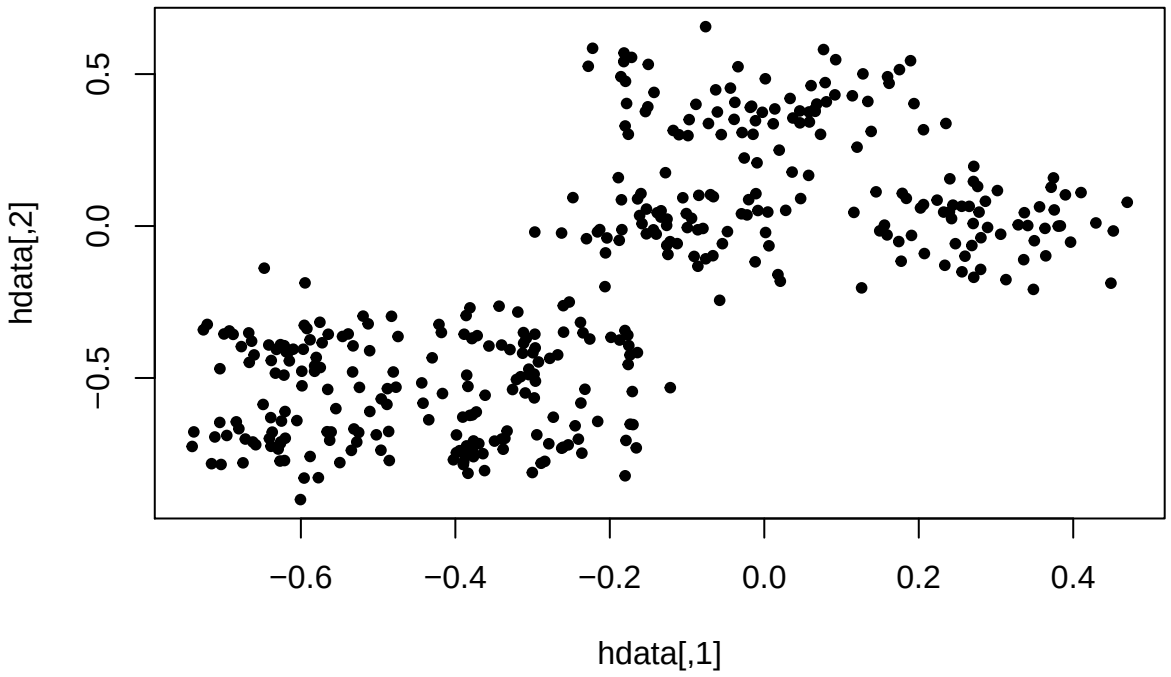
This result looks even better. NKE is no longer in a single group. Each group are well separated and the size difference is more moderate. This might because the Spearman correlation is invariant to monotonic transformation and thus is more robust to outliers. Therefore NKE is not separated in a single group.

2

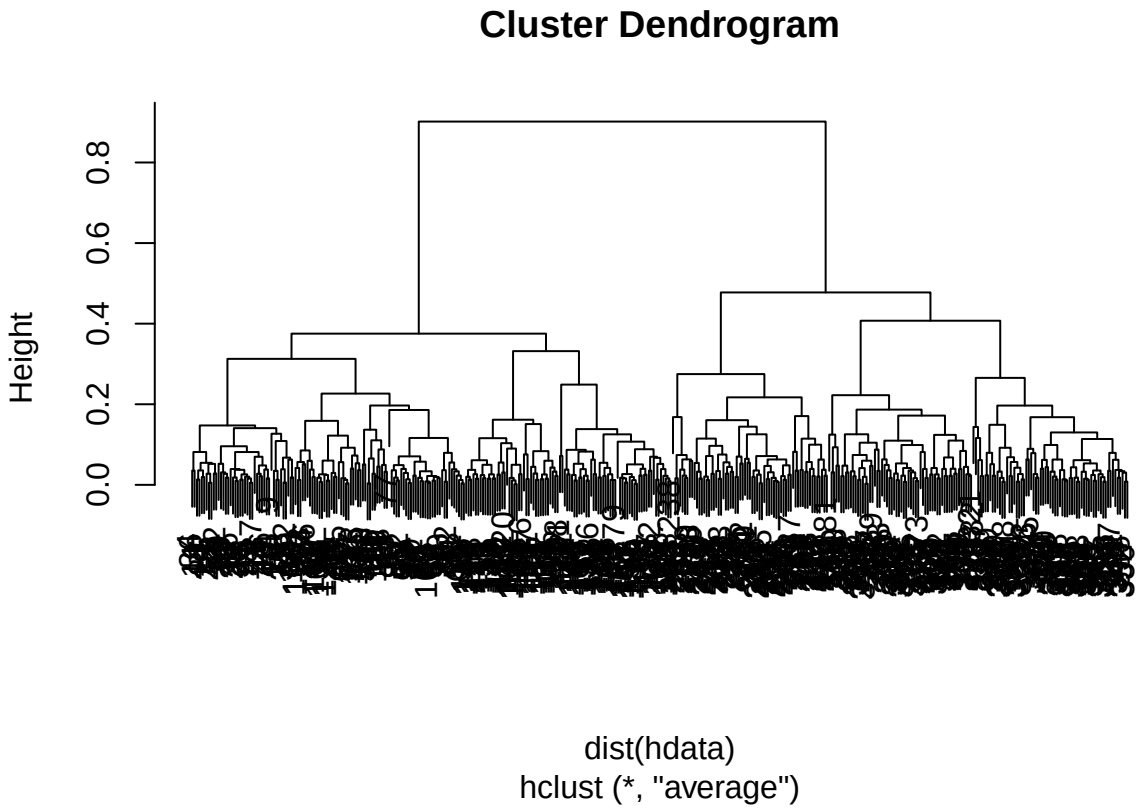
```
rm(list=ls())
load("hw5hierdata.Rdata")
```

(a)

```
plot(hdata, pch=20)
```




```
hold_clust5 = hclust(dist(hdata), method="average")
plot(hold_clust5)
```

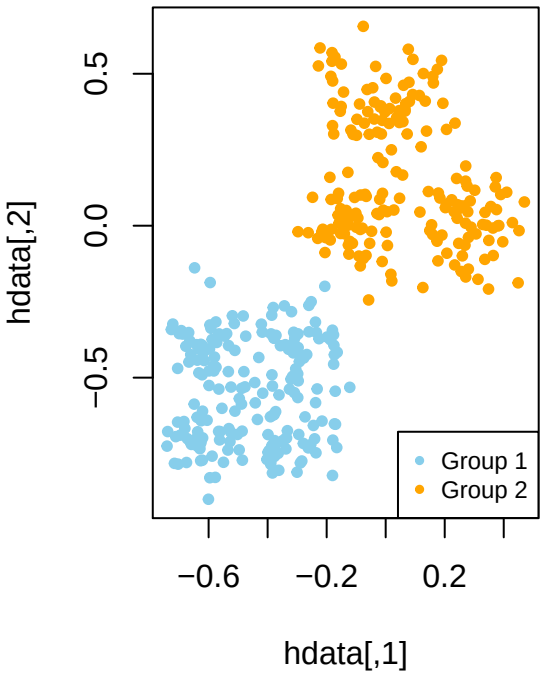


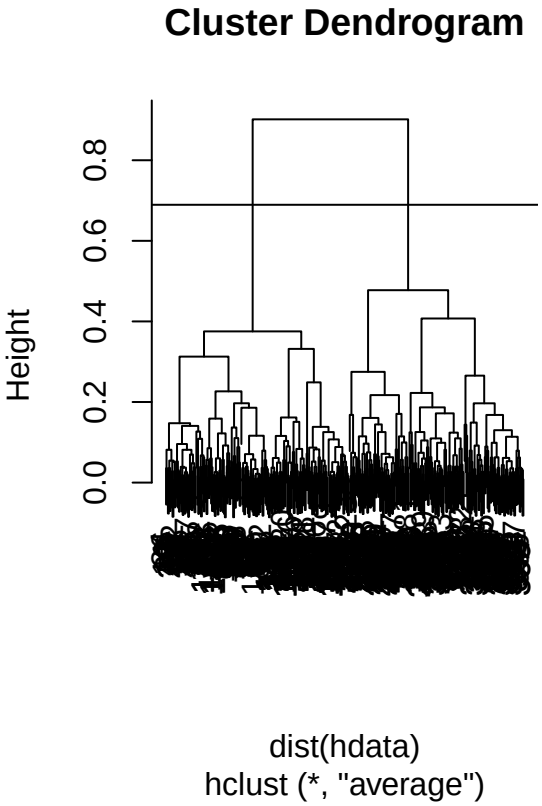
(b)

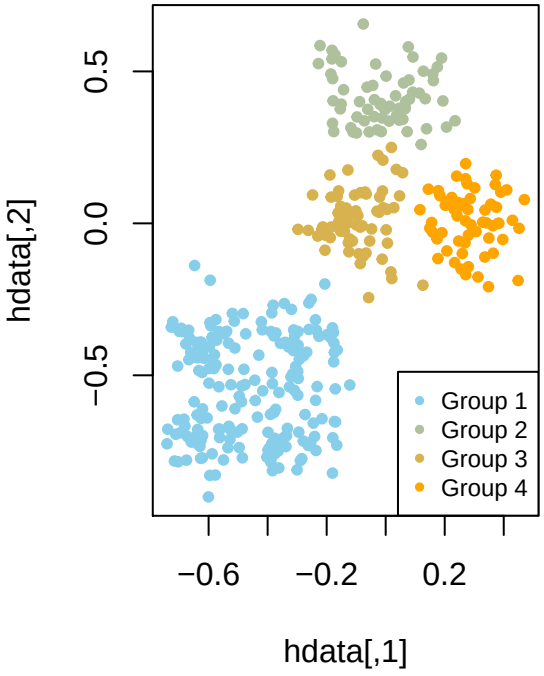
Big group 1 consists of the 4 subgroups on the left. Big group 2 consists of the 3 subgroups on the right.

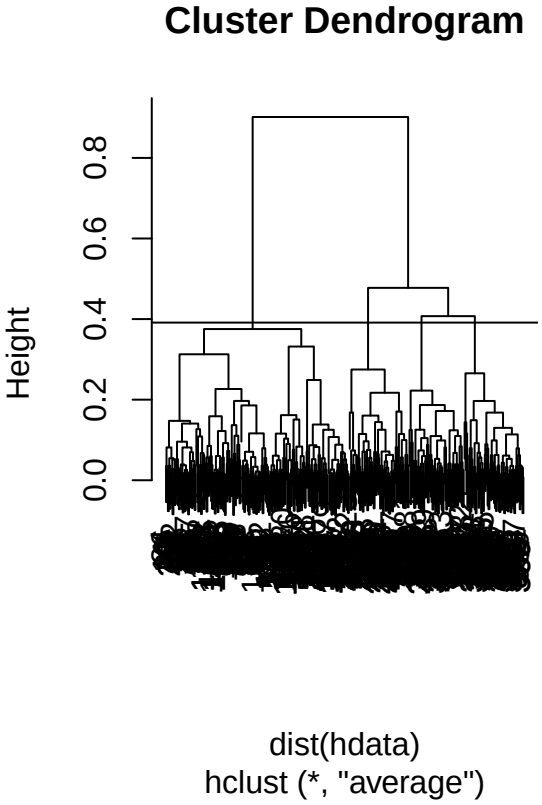
(c)

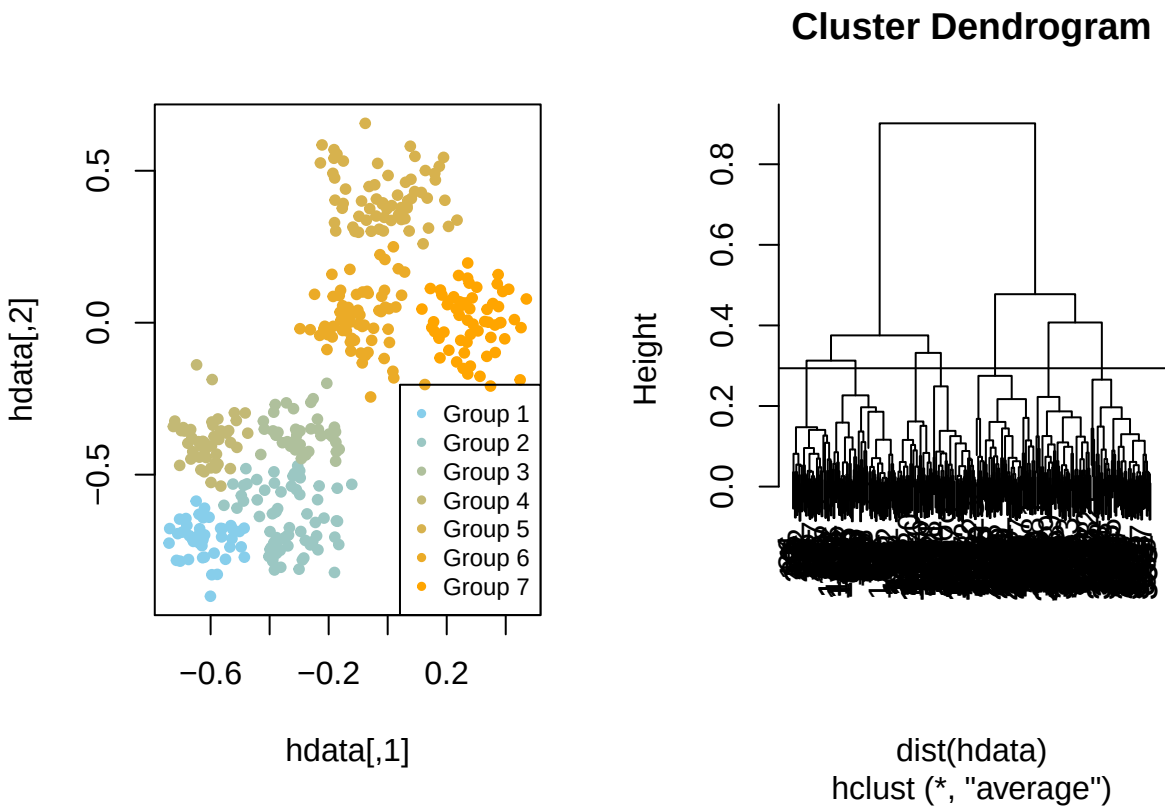
```
for (k in c(2,4,7)) {
  hold_cut5 = cutree(hold_clust5, k=k)
  tmp_palette = colorRampPalette(c('skyblue','orange'))(k)
  tmp_colors = tmp_palette[hold_cut5]
  par(mfrow=c(1,2))
  plot(hdata, pch=20, col=tmp_colors)
  legend("bottomright", legend=paste("Group", 1:k), pch=20, col=tmp_palette, cex=0.75)
  plot(hold_clust5)
  abline(h=mean(rev(hold_clust5$height)[(k-1):k]))
}
```


hdata[,2]
0.5
0.0
−0.5
−0.6
−0.2
0.2
hdata[,1]
Group 1
Group 2


Cluster Dendrogram
Height
0.0
0.2
0.4
0.6
0.8
dist(hdata)
hclust (*, "average")


hdata[,2]
0.5
0.0
−0.5
−0.6
−0.2
0.2
hdata[,1]
Group 1
Group 2
Group 3
Group 4

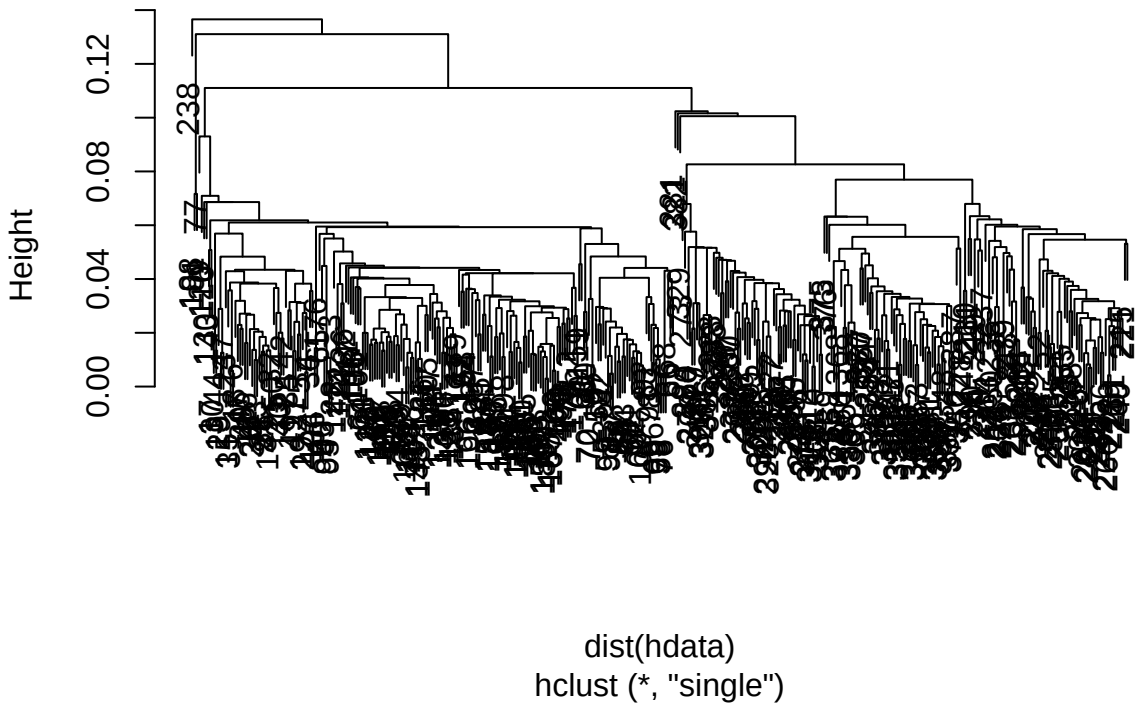

Cluster Dendrogram
Height
0.0
0.2
0.4
0.6
0.8
dist(hdata)
hclust (*, "average")



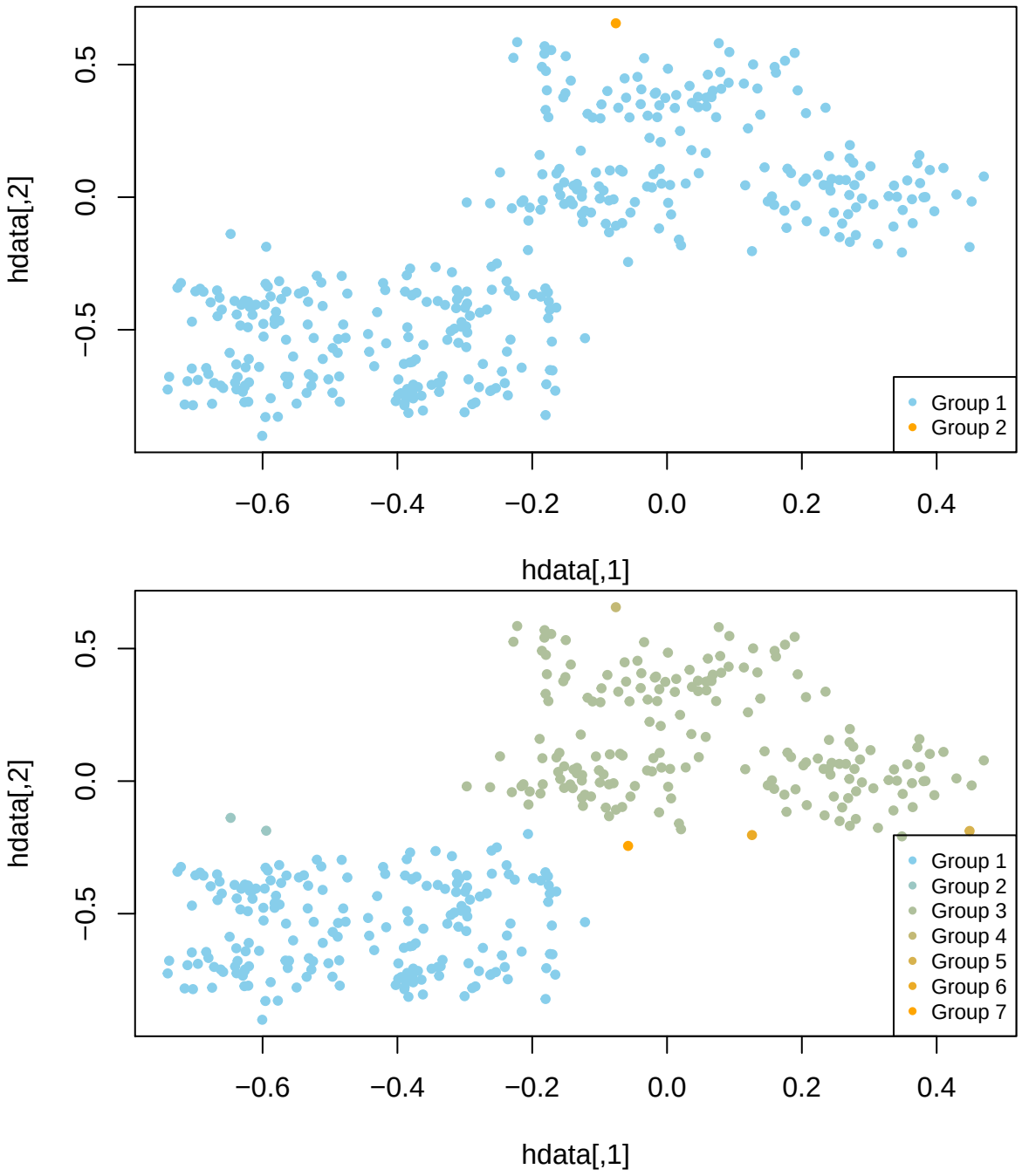
Hierarchial clustering correctly idendifies all the groups and subgroups. The result agrees with the labeling in (b).

(d)

```
hold_clust6 = hclust(dist(hdata), method="single")
plot(hold_clust6)
```




```
for (k in c(2,7)) {
  hold_cut6 = cutree(hold_clust6, k=k)
  tmp_palette = colorRampPalette(c('skyblue','orange'))(k)
  tmp_colors = tmp_palette[hold_cut6]
  plot(hdata, pch=20, col=tmp_colors)
  legend("bottomright", legend=paste("Group", 1:k), pch=20, col=tmp_palette, cex=0.75)
}
```



The clustering is not as clear as the previous one. We cannot easily separate big groups and subgroups from the dendrogram.

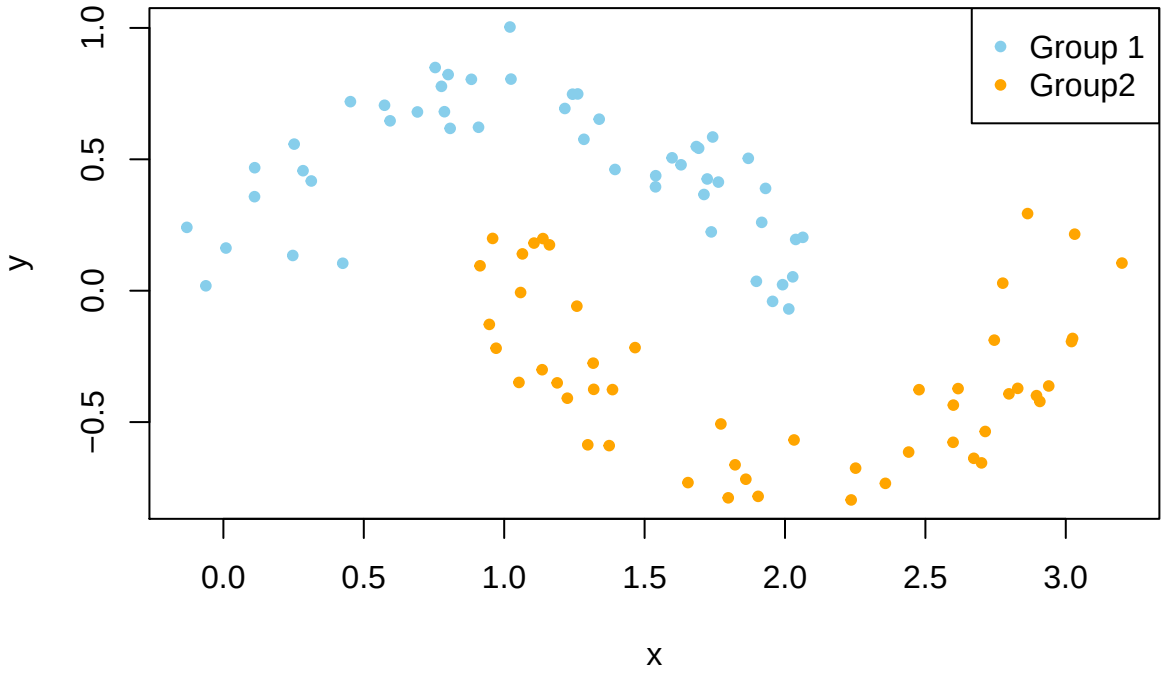
After cutting the dendrogram with the same number of clusters, the result is still not as clear as the previous one. The new clustering looks bad. This might because single linkage suffers from chaining. If a pair of points are close enough to each other, the two subgroups may merge into a big group, irrespective of the rest. The clusters formed under single linkage could be quite spread out.

3

```
rm(list=ls())
load("hw5single.Rdata")
```

(a)

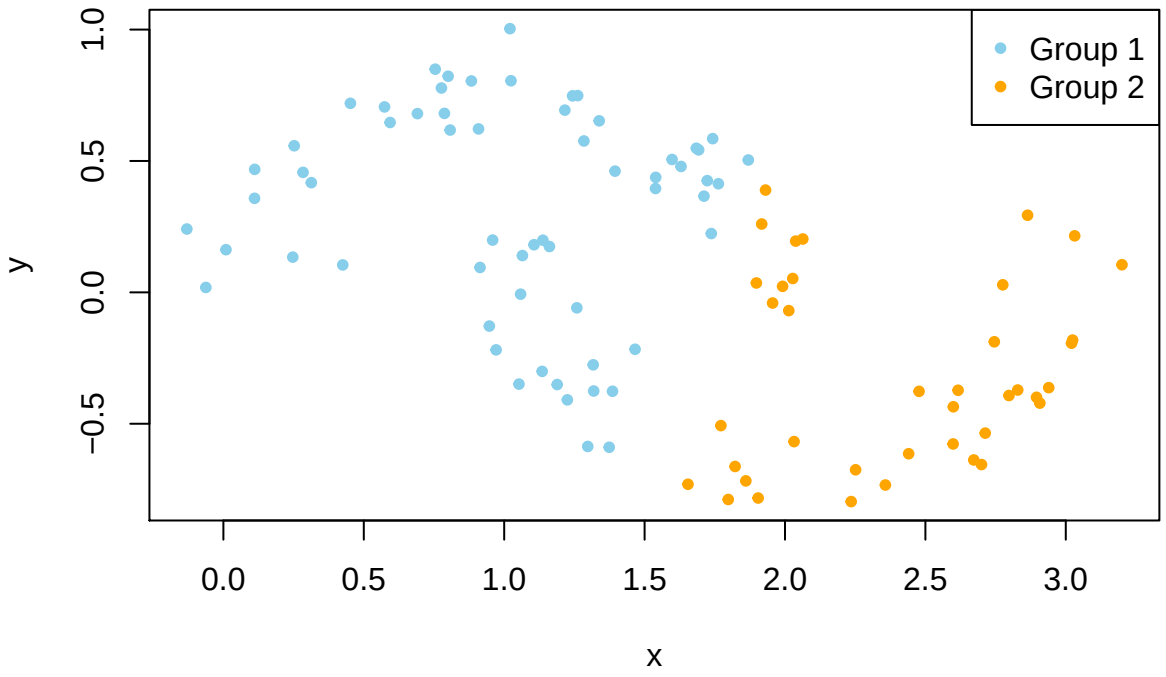
```
palette = colorRampPalette(c('skyblue','orange'))(2)
colors = palette[pieces]
plot(points, pch=20, col=colors)
legend("topright", legend=c("Group 1", "Group2"), pch=20, col=palette)
```



The cluster structure is very clear.

(b)

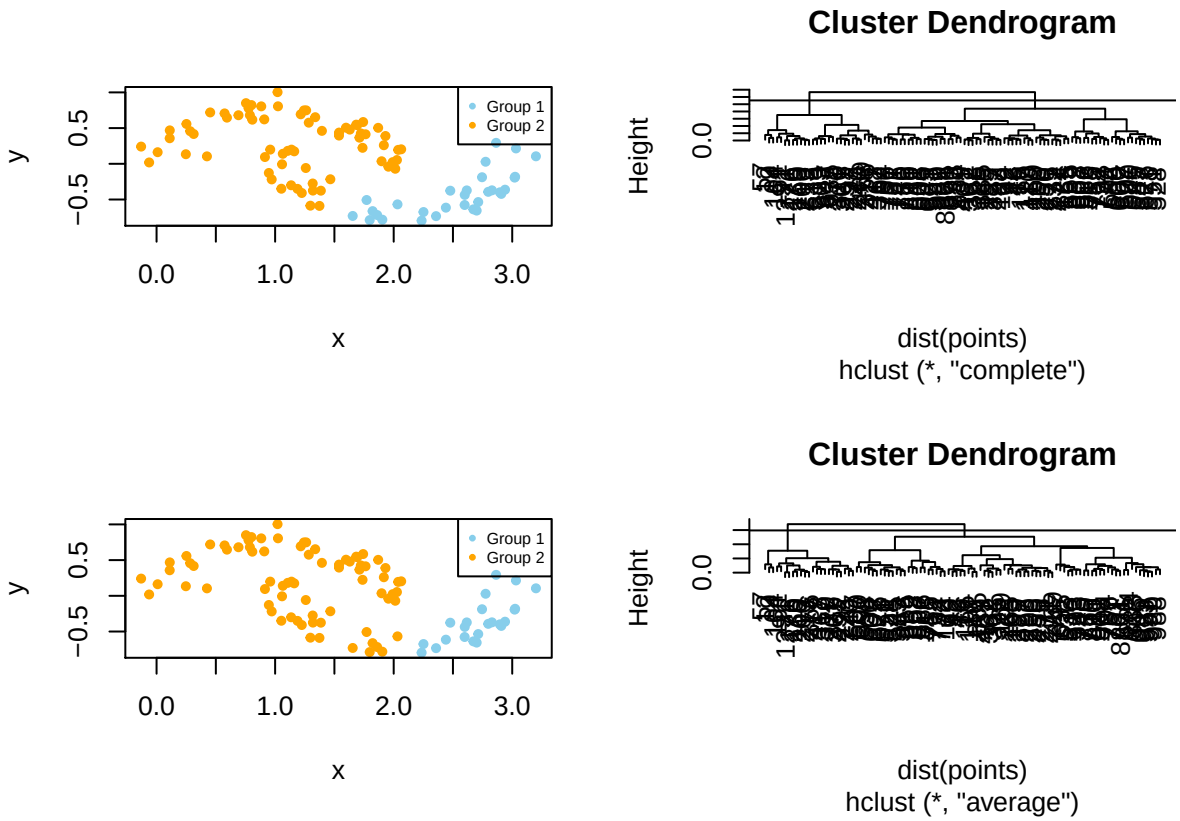
```
hold_kmeans = kmeans(points, centers=2)
colors2 = palette[hold_kmeans$cluster]
plot(points, pch=20, col=colors2)
legend("topright", legend=c("Group 1", "Group 2"), pch=20, col=palette)
```



K-means with 2 clusters works poorly here. It simply assigns the points on the left to one group and points on the right to another. This is a typical case of chaining: the two groups are quite spread out, with some pairs being very close to each other. The K-means method tries to assgin each point to the closest center, thus assigning outlier points to another group.

(c)

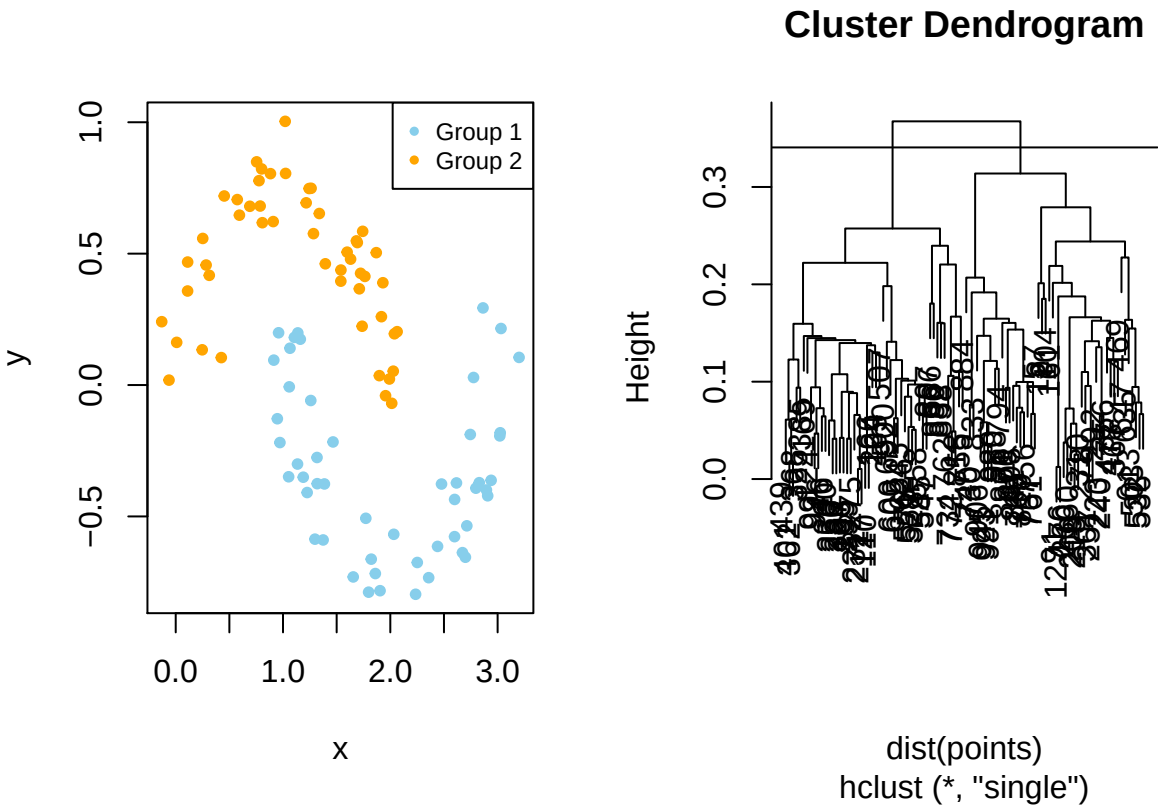
```
hold_clust7 = hclust(dist(points), method="complete")
hold_cut7 = cutree(hold_clust7, k=2)
hold_clust8 = hclust(dist(points), method="average")
hold_cut8 = cutree(hold_clust8, k=2)
par(mfrow=c(2,2))
plot(points, pch=20, col=palette[hold_cut7])
legend("topright", legend=c("Group 1", "Group 2"), pch=20, col=palette, cex=0.6)
plot(hold_clust7)
abline(h=mean(rev(hold_clust7$height)[1:2]))
plot(points, pch=20, col=palette[hold_cut8])
legend("topright", legend=c("Group 1", "Group 2"), pch=20, col=palette, cex=0.6)
plot(hold_clust8)
abline(h=mean(rev(hold_clust8$height)[1:2]))
```



The result is similar to the K-means method. Thw 2 groups are not correctly separated. This is because the complete linkage focuses on the distance of furthest pairs and average linkage considers all pairs, and the resulting clusters tend to be compact and not reflecting the true shapes of the 2 groups.

(d)

```
hold_clust9 = hclust(dist(points), method="single")
hold_cut9 = cutree(hold_clust9, k=2)
par(mfrow=c(1,2))
plot(points, pch=20, col=palette[hold_cut9])
legend("topright", legend=c("Group 1", "Group 2"), pch=20, col=palette, cex=0.75)
plot(hold_clust9)
abline(h=mean(rev(hold_clust9$height)[1:2]))
```



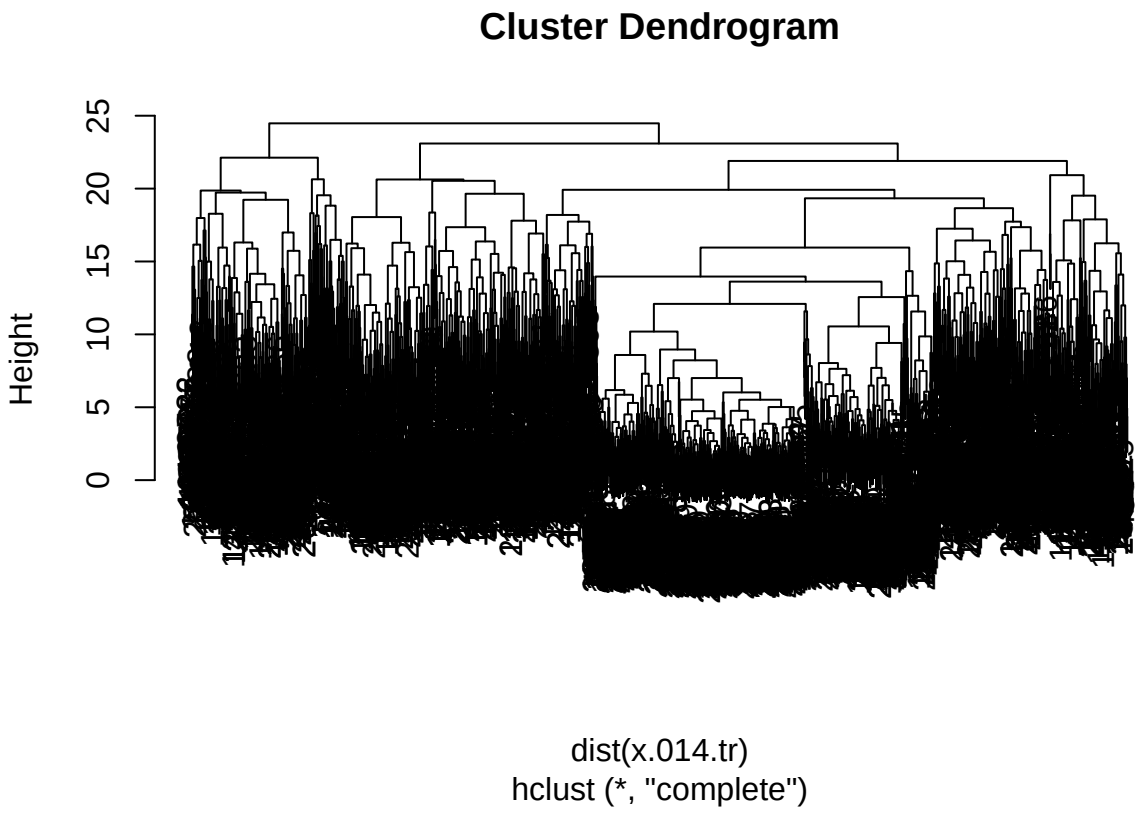
The single linkage works well, correctly identifying all points. This is because of the chaining of data. Single linkage only conisder the shortest distance pairs, so the groups can be quite spread out, as in this case.

4

```
load("zip.014.Rdata")
load("tangent_distances.RData")
source("plot.digit.R")
```

(a)

```
hold_clust10 = hclust(dist(x.014.tr), method="complete")
plot(hold_clust10)
```




```
table(cutree(hold_clust10,k=3), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1  373    0   95
##   2  214 1005  555
##   3  607    0    2
```

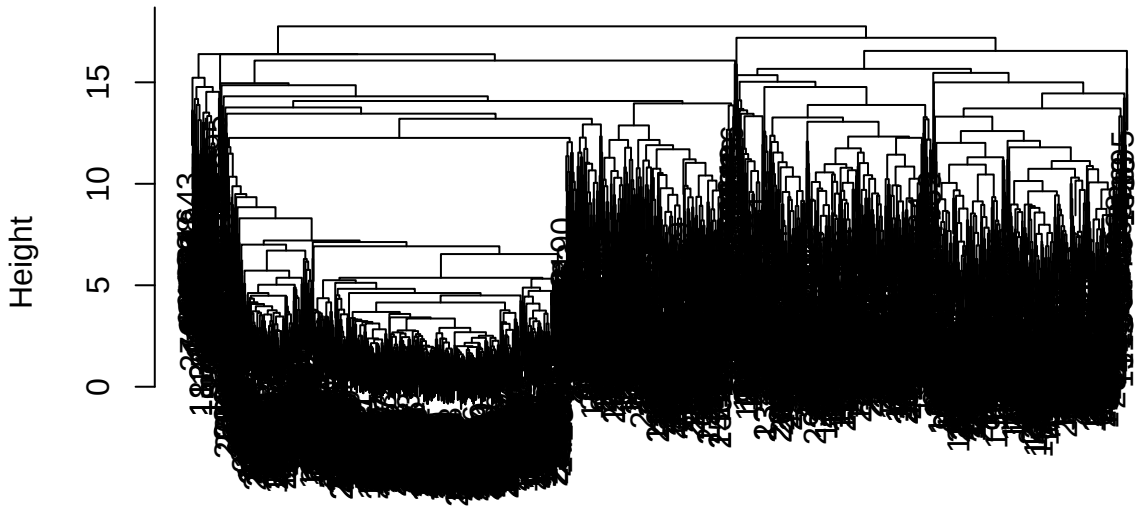
```
table(cutree(hold_clust10,k=6), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1   18    0   94
##   2    0 1005  532
##   3  607    0    2
##   4  355    0    1
##   5  211    0   23
##   6    3    0    0
```

Complete linkage works poorly, and that with 6 groups works slightly better. But either of them candistinguish 1 from 4 well, and groups 3~6 cannot differentiate.

(b)

```
hold_clust11 = hclust(dist(x.014.tr), method="average")
plot(hold_clust11)
```




```
table(cutree(hold_clust11,k=3), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1    3 1005  650
##   2 1188    0    2
##   3    3    0    0
```

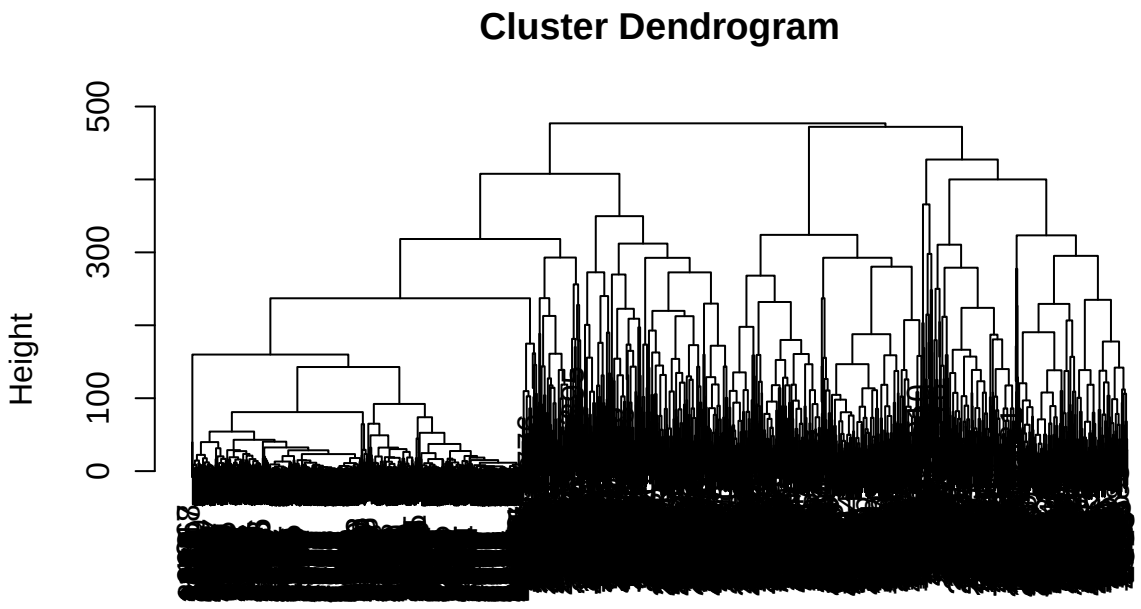
```
table(cutree(hold_clust11,k=6), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1    0    0   85
##   2    3 1005  564
##   3 1187    0    0
##   4    3    0    0
##   5    1    0    2
##   6    0    0    1
```

Average linkage with 6 groups does a better job separating 0 from the other two, but still cannot differentiate 1 from 4.

(c)

```
hold_clust12 = hclust(tangentDist, method="complete")
plot(hold_clust12)
```




```
table(cutree(hold_clust12,k=3), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1    0 1005  631
##   2  589    0    0
##   3  605    0   21
```

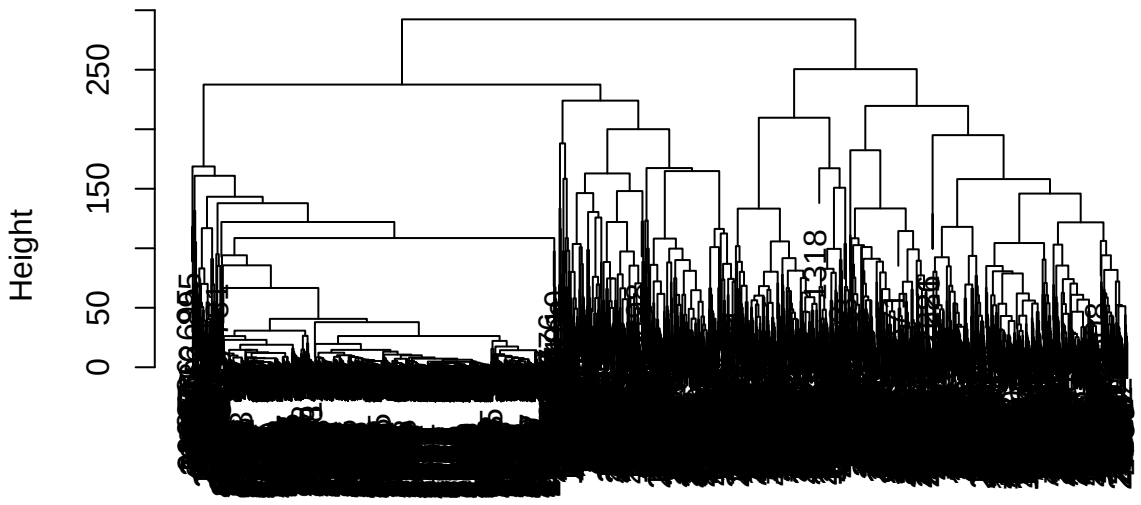
```
table(cutree(hold_clust12,k=6), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1    0    0  441
##   2    0 1005  190
##   3  589    0    0
##   4  341    0    0
##   5  250    0    1
##   6   14    0   20
```

Complete linkage with 6 groups is much better. It can differentiare 1 and 4 in most cases.

(d)

```
hold_clust13 = protoclust(tangentDist)
plot(hold_clust13)
```




```
table(cutree(hold_clust13,k=3), y.014.tr)
```

```
##    y.014.tr
##        0    1    4
##   1    1 1005  651
##   2  347    0    1
##   3  846    0    0
```

```
table(cutree(hold_clust13,k=6), y.014.tr)
```

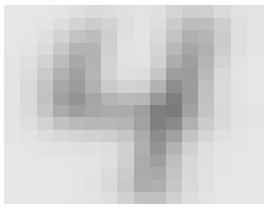
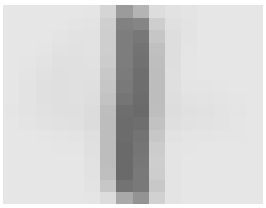
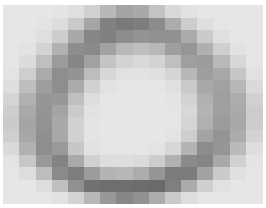
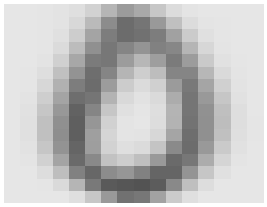
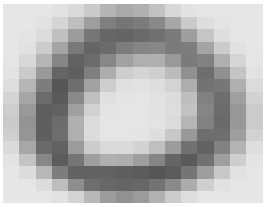
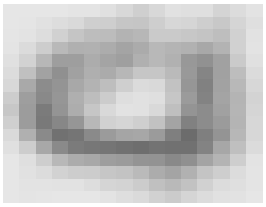
```
##    y.014.tr
##        0    1    4
##   1    0    0  505
##   2    1 1005  113
##   3  347    0    1
##   4  595    0    0
##   5    0    0   33
##   6  251    0    0
```

This have very similar results as in (c). The clustering with 6 groups is much better than using 3 groups. And it can further separate 1 from 4 compared with other methods.

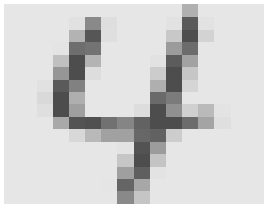
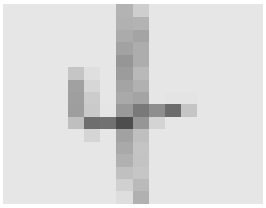
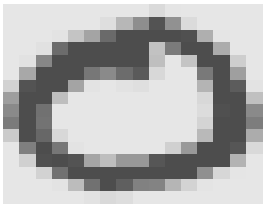
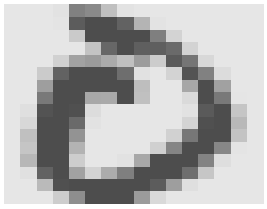
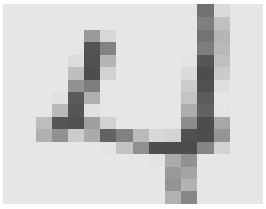
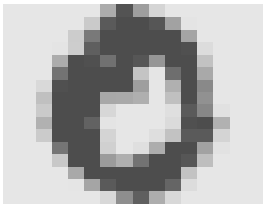
(e)

```
hold_cut12 = cutree(hold_clust12, k=6)
hold_cut13 = protocut(hold_clust13, k=6)
par(mfrow=c(2,3))
for (i in 1:6) {
```

```
  plot.digit(colMeans(x.014.tr[hold_cut12==i,]))
}
```

```
par(mfrow=c(2,3))
for (i in 1:6) {
  plot.digit(x.014.tr[hold_cut13$protos[i],])
}
```

The seond group of plots is easier to understand. Each plot is an actual image from the original dataset rather than a blurred average of images.