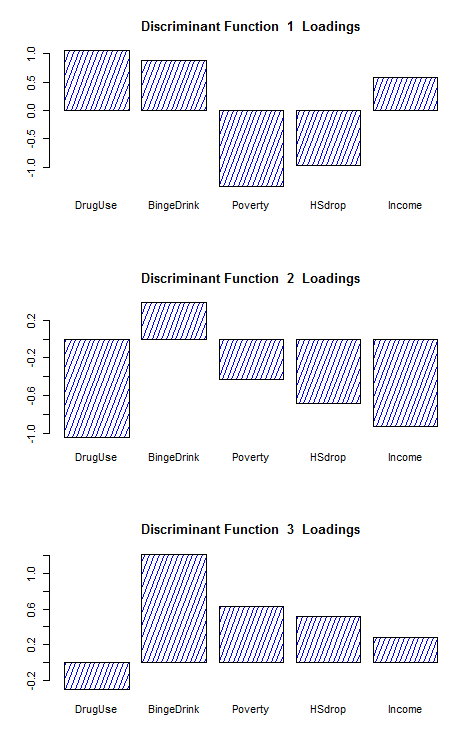
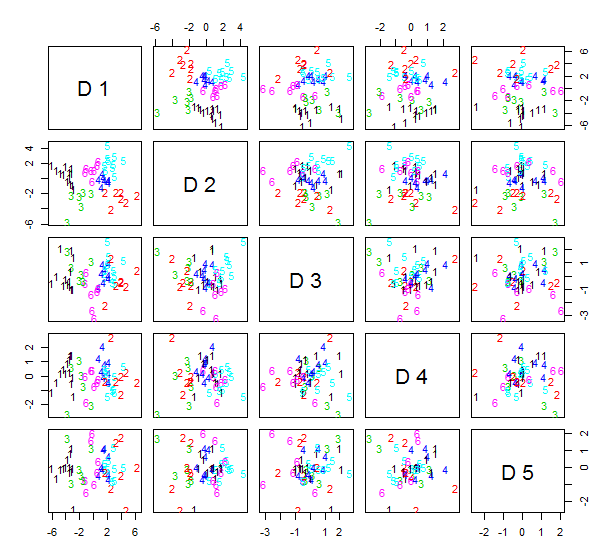
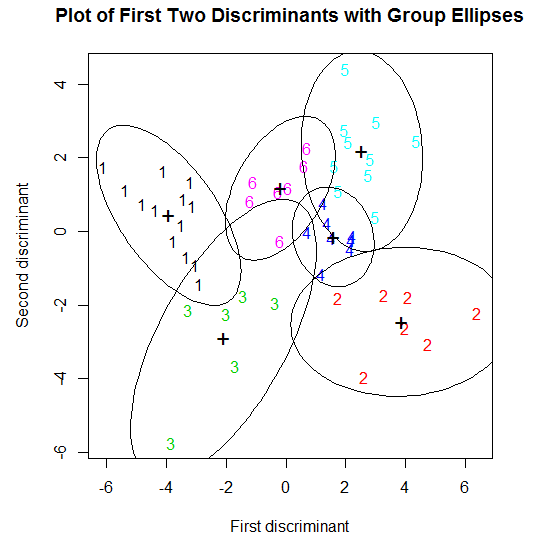
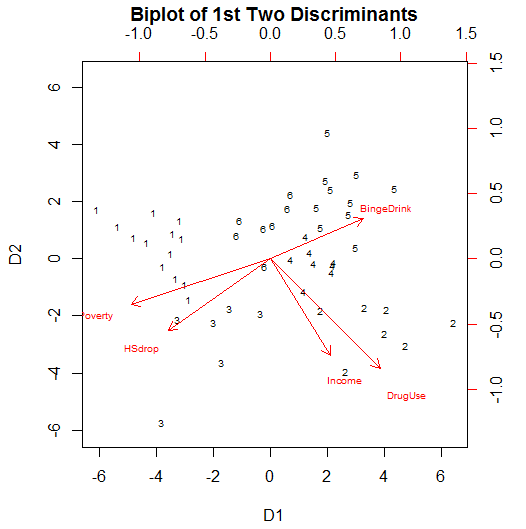
**5.2 – Exploring Clusters**

I have written a function called Discrim that performs a discriminant analysis using the clusters returned from a cluster analysis. We will discuss discriminant analysis later in the course, however given what we know about principal components we should be able to make sense of the results of this function as they are interpreted similarly. The Discrim function uses *linear discriminant analysis* to try to discriminate between cluster groups. Linear discriminant analysis finds linear combinations of the variables (think PCA) to try to maximally separate the identified groups (clusters here) in a data set. The results of this function give us some idea about how distinct our clusters are and what defining characteristics the clusters have.   
  
> Discrim(X,drug.clust)

The code for the Discrim function is given below:

|  |
| --- |
| Custom R Function |
| Discrim <- function (m, group, k = length(unique(group)), bar = T, pair = T, ell = T)  { par(ask = T)  par(err = 0)  m <- scale(as.matrix(m))  gp <- group  group <- as.numeric(as.factor(group))  m.disc <- lda(m, group)  disc.loads <- as.matrix(m.disc$scaling)  tol <- 1e-05  sig <- dim(m.disc$scaling)[2]  par(pty = "m")  par(mfrow = c(min(sig, 3), 1))  for (i in 1:min(sig, 3)) {  barplot(disc.loads[, i], density = 20, col = "blue")  title(main = paste("Discriminant Function ", i, " Loadings"),  cex = 0.6)  }  par(pty = "m")  par(mfrow = c(1, 1))  if (sig == 1) {  pair <- F  }  Dmat <- m %\*% disc.loads  dimnames(Dmat)[[2]] <- paste("D", 1:sig)  if (pair) {  pairs(Dmat, panel = function(x,y) {  par(err = -1)  gp = as.factor(group)  text(x, y, as.character(gp), col = as.numeric(gp))  invisible()  })  }  par(pty = "s")  par(mfrow = c(1, 1))  D1 <- m %\*% disc.loads[, 1]  D2 <- m %\*% disc.loads[, 2]  plot(D1, D2, type = "n", xlab = "First discriminant", ylab = "Second discriminant")  text(D1, D2, gp, col = as.numeric(group))  DD1 <- cbind(D1, group)  DD2 <- cbind(D2, group)  if (ell) {  for (i in 1:k) {  DD1temp <- DD1[group == i, ]  DD2temp <- DD2[group == i, ]  x <- DD1temp[, -2]  y <- DD2temp[, -2]  par(err = -1)  theta <- seq(0, 2 \* pi, len = 100)  j1 <- sin(theta)  j2 <- cos(theta)  xtemp <- cbind(x, y)  S <- var(xtemp)  eigS <- eigen(S)  P <- eigS$vectors  Lam <- diag(eigS$values)  sqrtS <- P %\*% Lam^0.5 %\*% t(P)  cons2 <- qchisq(0.95, 2)^0.5  junk <- cons2 \* t(sqrtS %\*% rbind(j1, j2))  xx <- junk[, 1] + mean(x)  yy <- junk[, 2] + mean(y)  lines(xx, yy)  points(mean(x), mean(y), pch = "+", cex = 1.5)  }  title(main = "Plot of First Two Discriminants with Group Ellipses",  cex = 0.7)  }  obs = as.matrix(cbind(D1, D2))  dloads = as.matrix(disc.loads[, 1:2])  biplot(x = obs, y = dloads, xlabs = as.character(group),  xlab = "D1", ylab = "D2", main = "Biplot of 1st Two Discriminants",  cex = 0.6)  invisible()  par(err = -1)  par(ask = F)  lda(m, gp)  } |

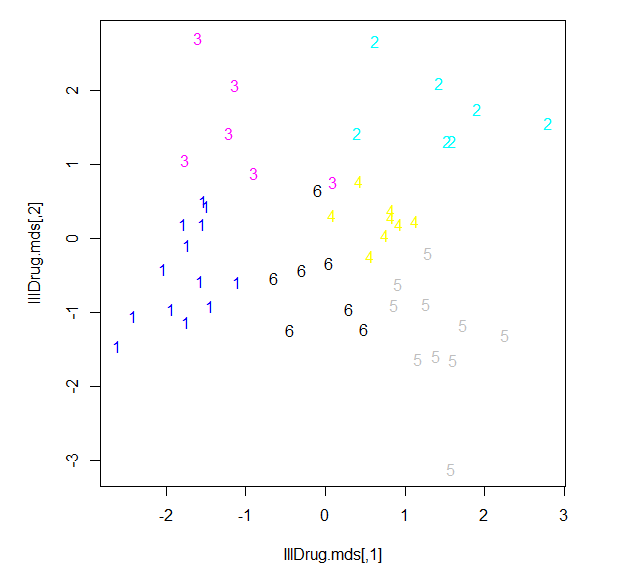
We can also label points in a multidimensional scaling of these data according to cluster membership.  
  
> library(MASS)

> X.dist = dist(X)

> IllDrug.mds = cmdscale(X.dist,k=2)

> plot(IllDrug.mds,type="n")

> text(IllDrug.mds,labels=as.character(drug.clust),col=as.numeric(drug.clust))



> IllDrug.iso = isoMDS(X.dist,k=2)

initial value 20.918363

iter 5 value 15.176204

iter 5 value 15.164617

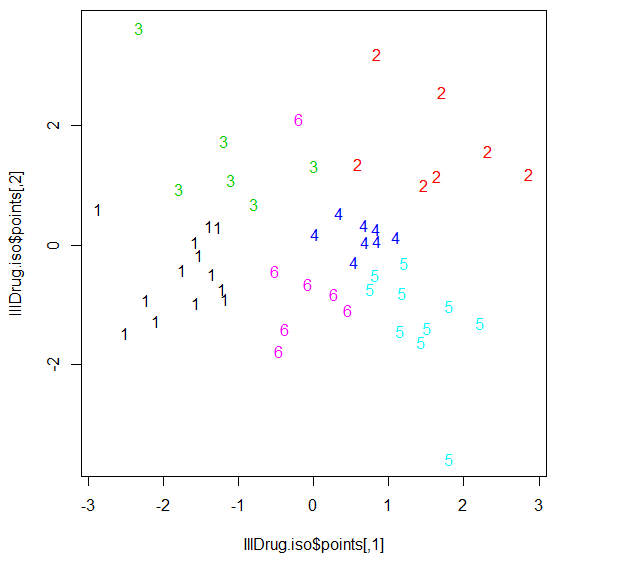
iter 5 value 15.164617

final value 15.164617

converged

> plot(IllDrug.iso$points,type="n")

> text(IllDrug.iso$points,labels=as.character(drug.clust),col=as.numeric(drug.clust))

> library(rgl)  
> IllDrug.iso = isoMDS(X.dist,k=3)

initial value 10.263657

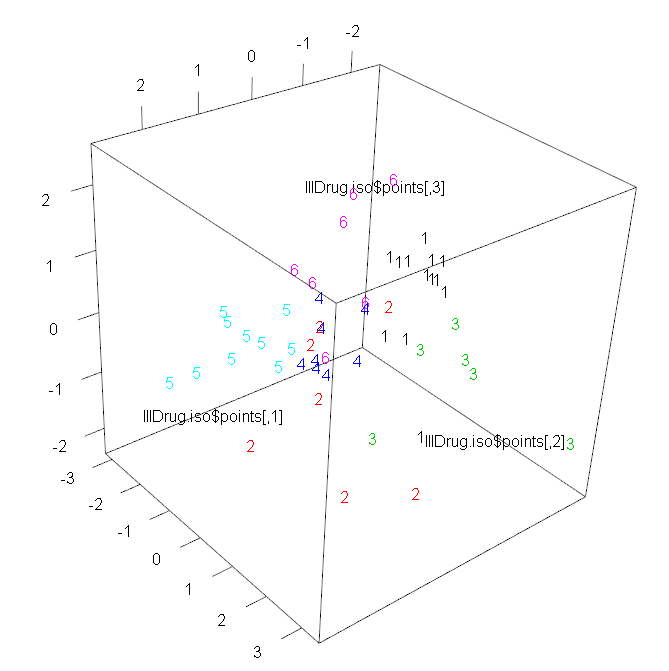
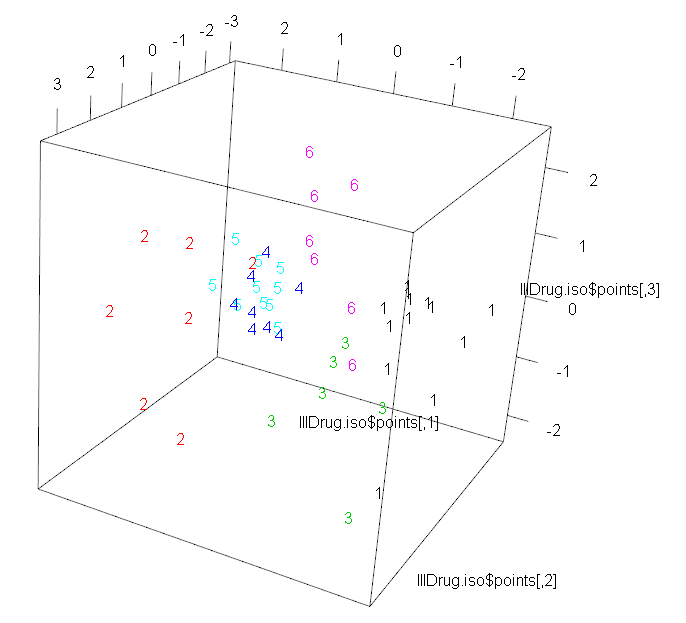
iter 5 value 7.096606

final value 6.988238

converged

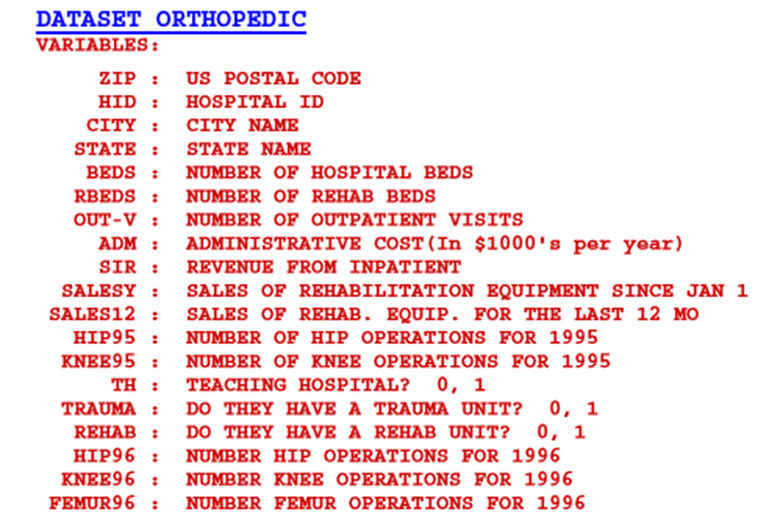
> plot3d(IllDrug.iso$points,type="n")

> text3d(IllDrug.iso$points,textss=as.character(drug.clust),col=as.numeric(drug.clust))



Example 5.3: Orthopedic Equipment Sales

These data were used in the multidimensional scaling notes (Section 3). Variable descriptions for these data are given below.



All of the numeric variables have been log transformed. For those with zeroes, was used.

> names(Orthopedic)

[1] "ZIP" "HospID" "City" "State" "Beds" "RBeds" "Outpatients"

[8] "Admin" "Inpatient" "Hip95" "Knee95" "SalesYr" "Sales12" "Teach"

[15] "Trauma" "Rehab" "Hip96" "Knee96" "Femur96" "logBeds" "logRBeds1"

[22] "logOut1" "logAdmin" "logInpat1" "lnHip951" "lnKnee951" "lnSales1" "lnSales121"

[29] "lnHip961" "lnKnee961" "lnFem961"

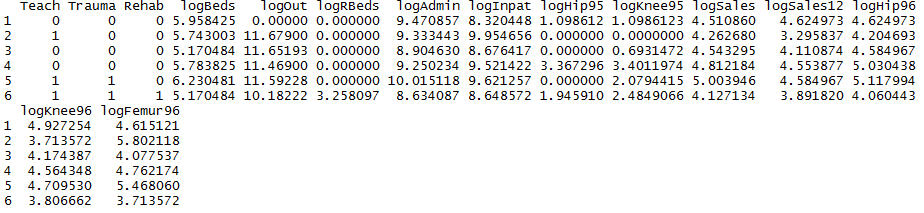
> ortho.mat = Orthopedic[,c(14:16,20:31)]

> names(ortho.mat)

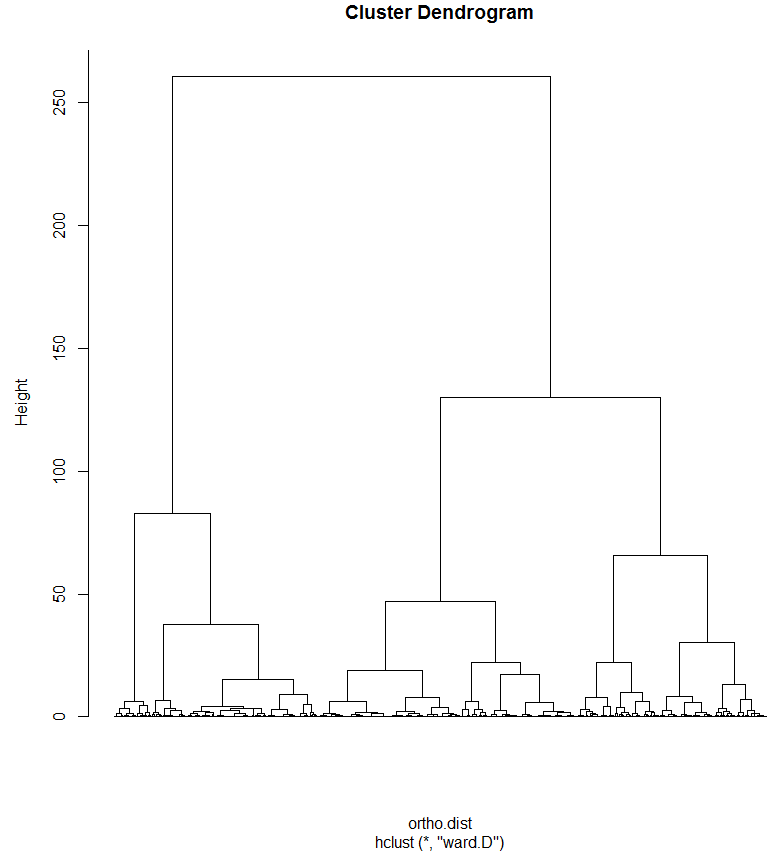
[1] "Teach" "Trauma" "Rehab" "logBeds" "logRBeds1" "logOut1" "logAdmin"

[8] "logInpat1" "lnHip951" "lnKnee951" "lnSales1" "lnSales121" "lnHip961" "lnKnee961"

[15] "lnFem961"

> head(ortho.mat)  


> ortho.dist = daisy(ortho.mat,type=list(asymm=c(1:3))) 🡨 Use Gower’s metric

> ortho.clust = hclust(ortho.dist,method=”ward.D”)  
> plot(ortho.clust,labels=FALSE,hang=-1)  
> grps = cutree(ortho.clust,k=10)  


> clust.grps(ortho.mat,grps,parcoord=T)

Cluster 1 consists of

=======================================================================

1 12 16 37 39 65 71 83 94 106 107 136 161 163 167 175 179 184 191 195 198 200 201 207 208 211 233 240 270 274 283 295 313 344 348 356 374 375 380 402 415 416 431 443 459 466 468 487 490 492 497 499 519 520 523 532 533 536 554 560 563 575 581 584 598 600 606 616 624 626 640 642 649 651 652 655 658 659 661 668 672 692 702 721 722 724 725 726 736 743 744 746 763 768 769 777 796 808 831 833 844 848 857 863 876 879 882 886 890 900 912 953 971 982 996 1007 1075 1082 1083 1111 1122 1127 1128 1186 1209 1234 1254 1276 1281 1290 1298 1308 1318 1321 1324 1330 1333 1335 1336 1356 1358 1371 1375 1380 1397 1409 1418 1437 1438 1441 1462 1495 1502 1512 1527 1532 1533 1554 1572 1581 1582 1586 1599 1612 1623 1630 1638 1640 1649 1652 1654 1655 1677 1687 1688 1689 1702 1703 1709 1713 1734 1737 1751 1754 1756 1758 1762 1763 1765 1767 1768 1770 1781 1782 1797 1801 1803 1804 1809 1810 1817 1818 1820 1825 1832 1837 1842 1843 1846 1848 1849 1891 1916 1919 1921 1922 1932 1982 2001 2013 2015 2054 2062 2074 2075 2078 2106 2110 2113 2123 2134 2150 2152 2184 2192 2205 2206 2209 2219 2224 2237 2253 2262 2263 2265 2268 2271 2275 2285 2295 2305 2308 2310 2311 2317 2318 2331 2349 2353 2359 2378 2393 2395 2415 2417 2420 2431 2433 2436 2441 2458 2459 2462 2471 2489 2490 2491 2499 2502 2505 2508 2510 2514 2515 2516 2520 2527 2529 2533 2543 2544 2547 2550 2551 2552 2553 2554 2563 2567 2568 2569 2583 2591 2593 2598 2614 2624 2629 2630 2636 2646 2656 2667 2668 2682 2684 2695 2697 2702 2707 2716 2720 2722 2733 2735 2736 2748 2770 2772 2783 2793 2795 2818 2819 2820 2825 2826 2827 2828 2834 2843 2860 2874 2883 2894 2895 2915 2922 2934 2935 2937 2940 2946 2947 2948 2954 2957 2964 2970 2979 2980 2981 2983 2984 2985 2987 2989 2990 2992 3005 3011 3012 3015 3018 3024 3028 3032 3035 3042 3047 3048 3051 3052 3053 3067 3069 3078 3087 3089 3090 3103 3105 3109 3115 3118 3120 3143 3150 3160 3165 3167 3168 3176 3180 3189 3195 3196 3197 3202 3207 3209 3211 3213 3215 3217 3219 3224 3231 3232 3237 3241 3245 3257 3260 3261 3268 3280 3282 3286 3288 3290 3295 3298 3299 3300 3305 3306 3320 3324 3339 3340 3347 3353 3367 3371 3377 3396 3397 3399 3401 3404 3413 3414 3435 3449 3482 3486 3489 3492 3494 3499 3504 3506 3510 3511 3513 3527 3529 3557 3558 3563 3584 3621 3649 3665 3678 3687 3716 3727 3748 3751 3752 3753 3754 3755 3772 3779 3780 3782 3787 3792 3793 3795 3796 3802 3808 3826 3846 3847 3875 3895 3896 3899 3904 3906 3908 3911 3913 3920 3924 3928 3932 3933 3937 3945 3946 3949 3956 3983 3984 3987 3988 3997 4002 4008 4009 4020 4056 4059 4061 4071 4078 4086 4088 4090 4111 4113 4115 4121 4124 4126 4128 4138 4152 4158 4166 4169 4177 4188 4191 4193 4205 4207 4209 4215 4220 4228 4233 4235 4237 4244 4252 4253 4254 4256 4257 4260 4263 4268 4269 4270 4273 4287 4288 4289 4298 4302 4306 4312 4317 4327 4330 4337 4338 4343 4346 4348 4357 4358 4363 4364 4366 4368 4376 4389 4395 4398 4404 4416 4417 4421 4430 4431 4434 4438 4443 4446 4448 4454 4459 4468 4469 4472 4473 4478 4482 4483 4486 4487 4488 4489 4490 4494 4495 4496 4499 4512 4516 4520 4524 4525 4526 4527 4533 4534 4536 4538 4540 4541 4543 4544 4552 4553 4573 4577 4579 4580 4589 4590 4592 4596 4605 4608 4610 4611 4621 4622 4624 4631 4634 4637 4638 4639 4643 4644 4646 4648 4649 4650 4652 4655 4661 4662 4668 4672 4673 4677 4678 4679 4681 4683 4686 4688 4689 4691 4696 4701

Variable means in this cluster are:

--------------------------------------------------

Teach Trauma Rehab logBeds logRBeds1 logOut1 logAdmin logInpat1 lnHip951 lnKnee951 lnSales1 lnSales121

0.0000000 0.0000000 0.0000000 2.0438232 0.0000000 3.2514276 3.5835799 3.4662752 0.9311017 1.0941619 1.4747648 1.2495588

lnHip961 lnKnee961 lnFem961

1.4940390 1.2796823 1.5363210

Cluster 2 consists of

=======================================================================

2 8 9 11 14 20 22 27 29 30 38 40 48 50 57 62 67 68 76 77 90 92 93 96 99 100 101 102 105 114 115 120 124 127 128 130 132 134 140 141 143 145 148 150 151 152 153 158 159 169 174 177 178 181 182 183 192 193 194 196 197 202 209 215 227 232 234 236 243 254 260 261 265 266 267 273 287 288 296 309 310 314 316 320 321 322 324 325 330 331 338 343 347 353 358 361 362 365 366 369 370 376 377 378 385 387 392 393 404 417 418 424 430 438 440 444 448 449 453 456 465 470 480 494 495 513 518 538 546 558 561 574 585 588 591 601 603 610 623 629 641 708 710 712 715 718 728 729 731 733 753 759 764 766 775 780 783 784 786 789 801 809 810 813 816 841 849 860 866 869 871 903 904 907 910 918 924 932 933 950 958 959 967 968 973 976 977 985 986 988 989 991 992 1001 1006 1009 1011 1012 1014 1019 1020 1021 1026 1031 1042 1045 1056 1059 1069 1071 1086 1096 1098 1112 1115 1117 1120 1121 1159 1195 1222 1227 1239 1241 1264 1282 1295 1296 1303 1304 1309 1312 1315 1317 1323 1325 1326 1328 1329 1332 1341 1344 1363 1367 1376 1377 1379 1385 1386 1393 1399 1403 1408 1425 1427 1428 1459 1467 1471 1475 1513 1520 1523 1535 1546 1555 1563 1564 1565 1566 1568 1571 1573 1580 1587 1593 1596 1604 1607 1619 1663 1666 1674 1678 1682 1686 1704 1722 1730 1749 1759 1769 1775 1778 1786 1787 1789 1792 1794 1814 1815 1822 1826 1840 1850 1853 1855 1867 1869 1878 1882 1883 1906 1920 1927 1935 1946 1950 1958 1959 1966 1967 1974 1976 1978 1989 1990 1995 1996 2004 2006 2009 2010 2023 2025 2032 2049 2056 2065 2069 2072 2077 2079 2082 2085 2091 2096 2102 2103 2114 2117 2118 2119 2128 2136 2148 2156 2171 2173 2174 2194 2197 2204 2216 2227 2233 2236 2238 2242 2246 2247 2251 2255 2297 2302 2315 2320 2326 2327 2334 2337 2339 2340 2342 2357 2362 2363 2366 2372 2375 2379 2384 2387 2391 2404 2411 2414 2418 2426 2428 2442 2447 2467 2469 2484 2561 2572 2575 2579 2584 2585 2587 2601 2602 2605 2606 2610 2616 2619 2660 2662 2670 2699 2704 2705 2734 2749 2750 2784 2790 2792 2833 2853 2875 2898 2920 2931 2945 2974 3008 3009 3027 3046 3060 3061 3063 3107 3124 3125 3141 3172 3187 3203 3206 3210 3216 3239 3240 3243 3247 3248 3249 3283 3333 3334 3335 3351 3352 3355 3376 3430 3437 3444 3445 3448 3461 3472 3473 3474 3481 3487 3496 3501 3508 3617 3639 3646 3653 3654 3659 3664 3674 3683 3697 3703 3710 3744 3745 3801 3843 3858 3859 3871 3909 3910 3912 3919 3921 3951 3957 3965 3968 4025 4051 4098 4104 4105 4116 4123 4125 4144 4146 4176 4203 4204 4222 4230 4266 4322 4353 4354 4405 4406 4408 4435 4437 4515 4531 4563 4567 4582 4595 4623 4628 4636 4641 4647 4651 4669 4682 4687

Variable means in this cluster are:

--------------------------------------------------

Teach Trauma Rehab logBeds logRBeds1 logOut1 logAdmin logInpat1 lnHip951 lnKnee951 lnSales1 lnSales121

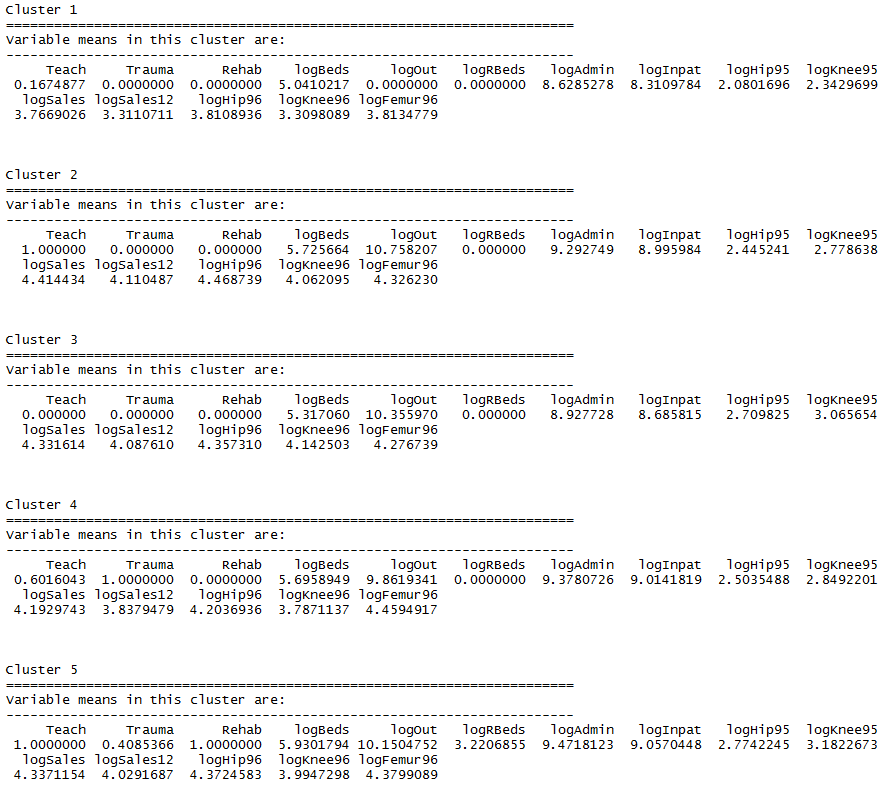
1.0000000 0.0000000 0.0000000 2.4429718 0.0000000 4.0783103 3.9752120 3.8077641 0.8711817 1.0005358 1.7632446 1.5987789

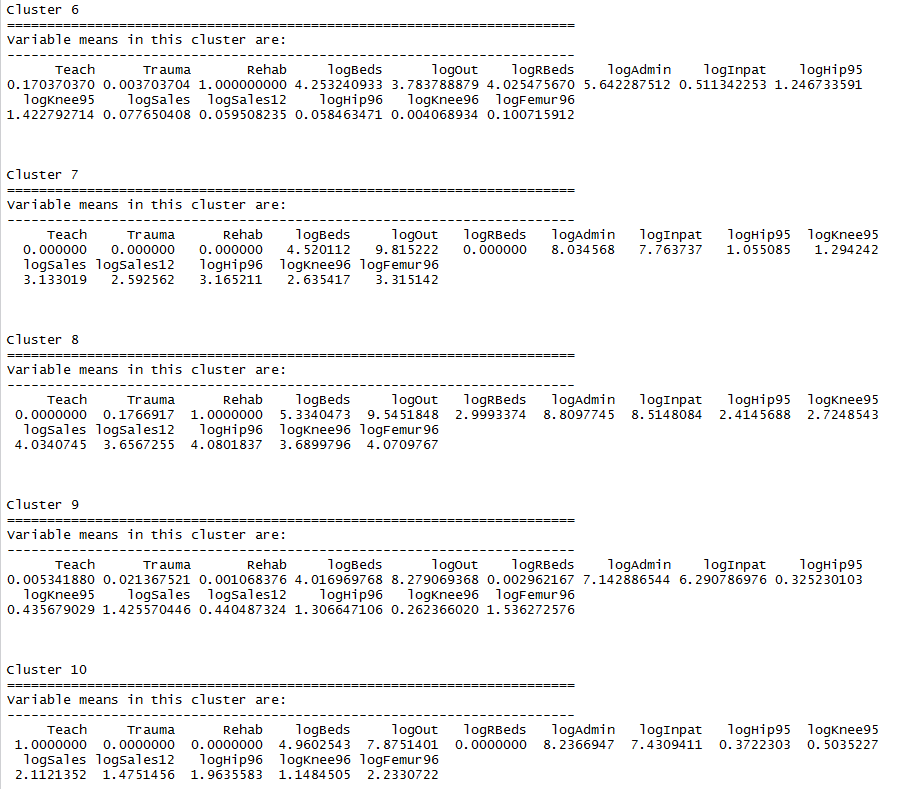
lnHip961 lnKnee961 lnFem961

1.7760076 1.5775709 1.7550238

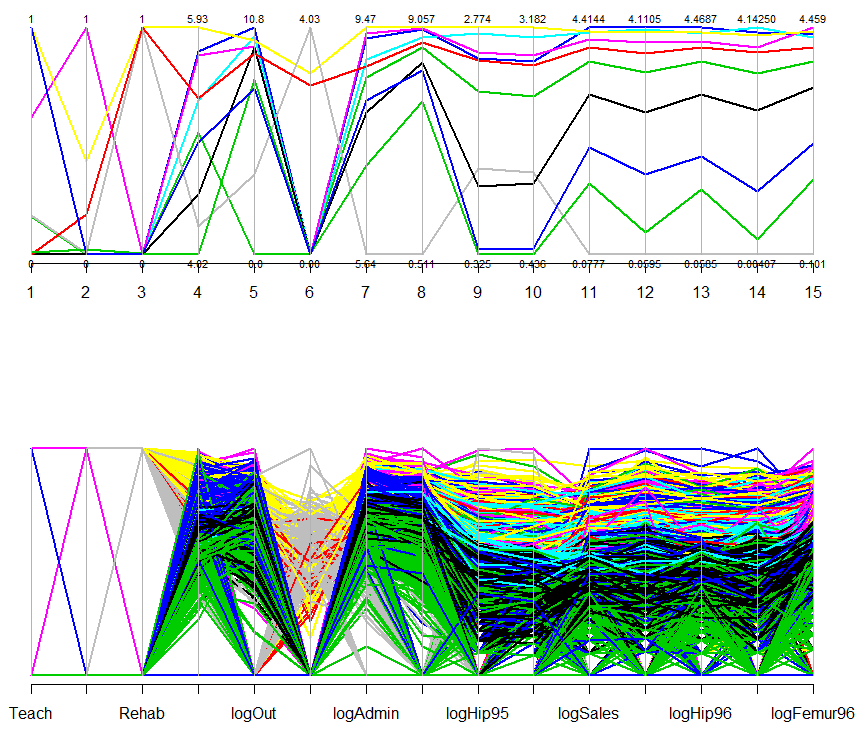
...   
YIKES! Too many observation labels to possibly digest! We can use the suppress=T option to suppress the observation labels from the cluster report.

> clust.grps(ortho.mat,grps,parcoord=T,suppress=T)





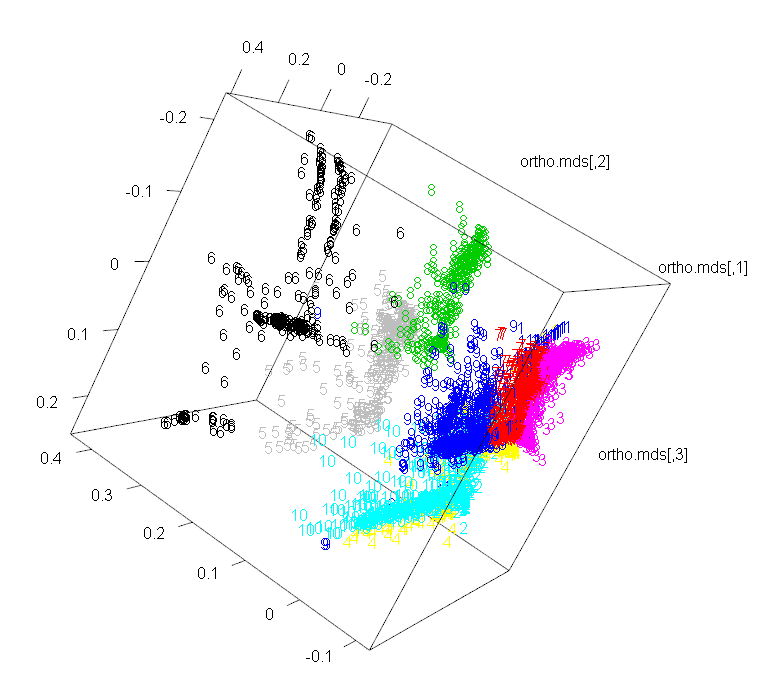
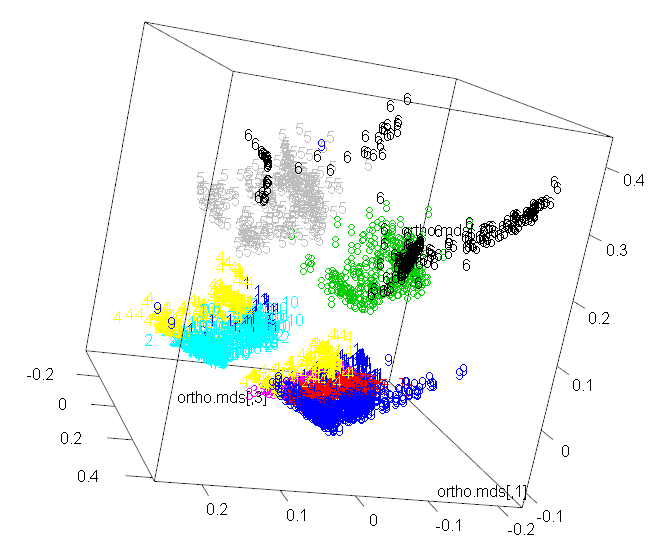
The parallel coordinate plots shows the mean variable values for each cluster and individuals below.

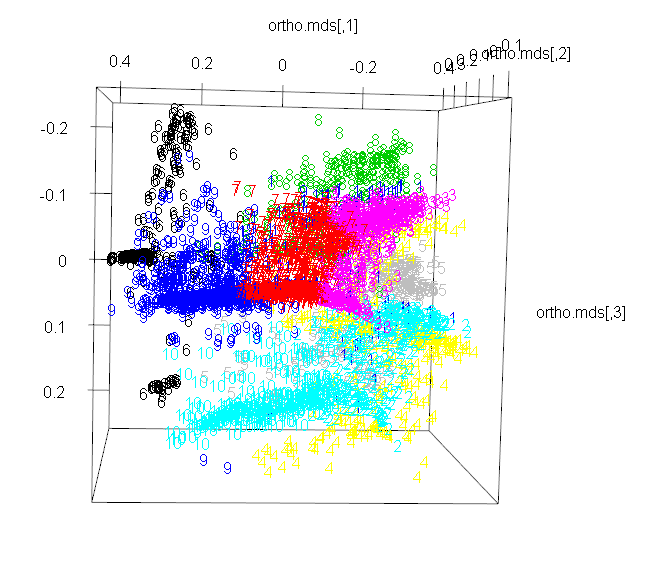
Again we use multidimensional scaling to view clusters.

> ortho.mds = cmdscale(ortho.dist,k=3)

> plot3d(ortho.mds,type="n")

> text3d(ortho.mds,texts=as.character(grps),col=as.numeric(grps)+3)





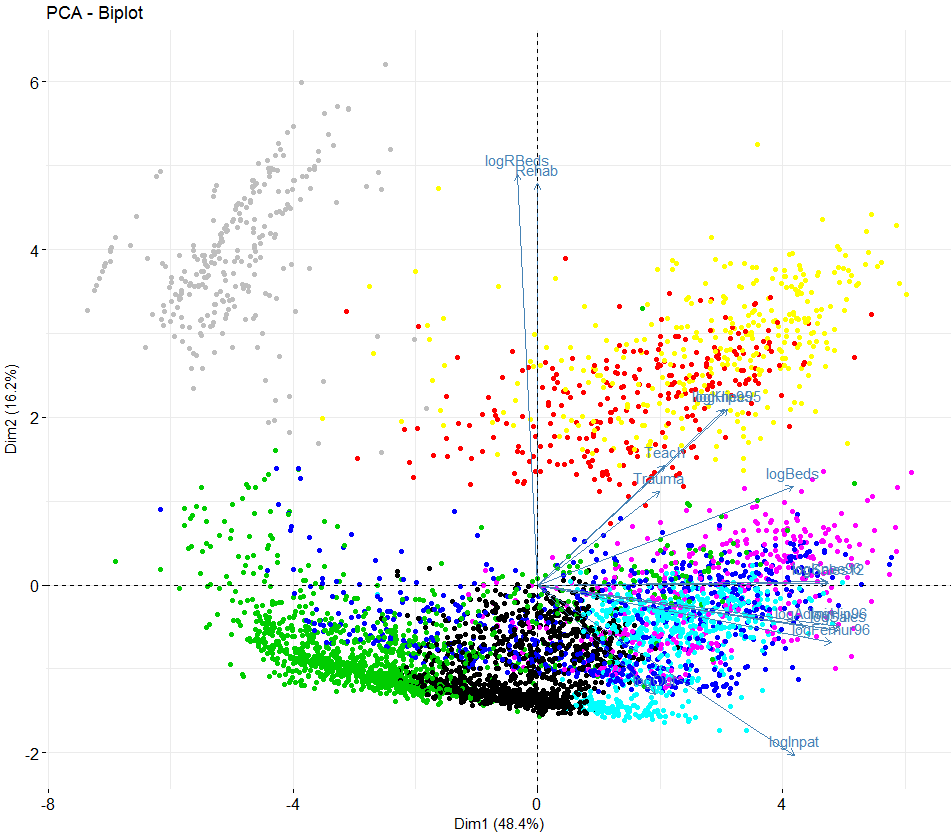
We could also use PCA to visualize these data in lower dimensional space. We will use the function fviz\_pca\_biplot in the factoextra package to display the biplot for PCA of these data. This version of the biplot function allows us to color code the points by a nominal grouping variable.

> ortho.PCA = prcomp(scale(ortho.mat))

> library(factoextra)

> fviz\_pca\_biplot(ortho.PCA,col.ind=as.numeric(grps)+2,label=”var”)

> fviz\_pca\_biplot(ortho.PCA,habillage=as.factor(grps),label=”var”)





**5.3 – The cluster Library and Other Hierarchical Clustering Methods**  
Example 5.2: Illicit drug use in the U.S. (cont’d)

The library cluster has more advanced methods of performing cluster analysis as well functions for assessing the “goodness” of the clustering.

Agglomerative Clustering (agnes) – alternative to hclust

> names(IllDrug)

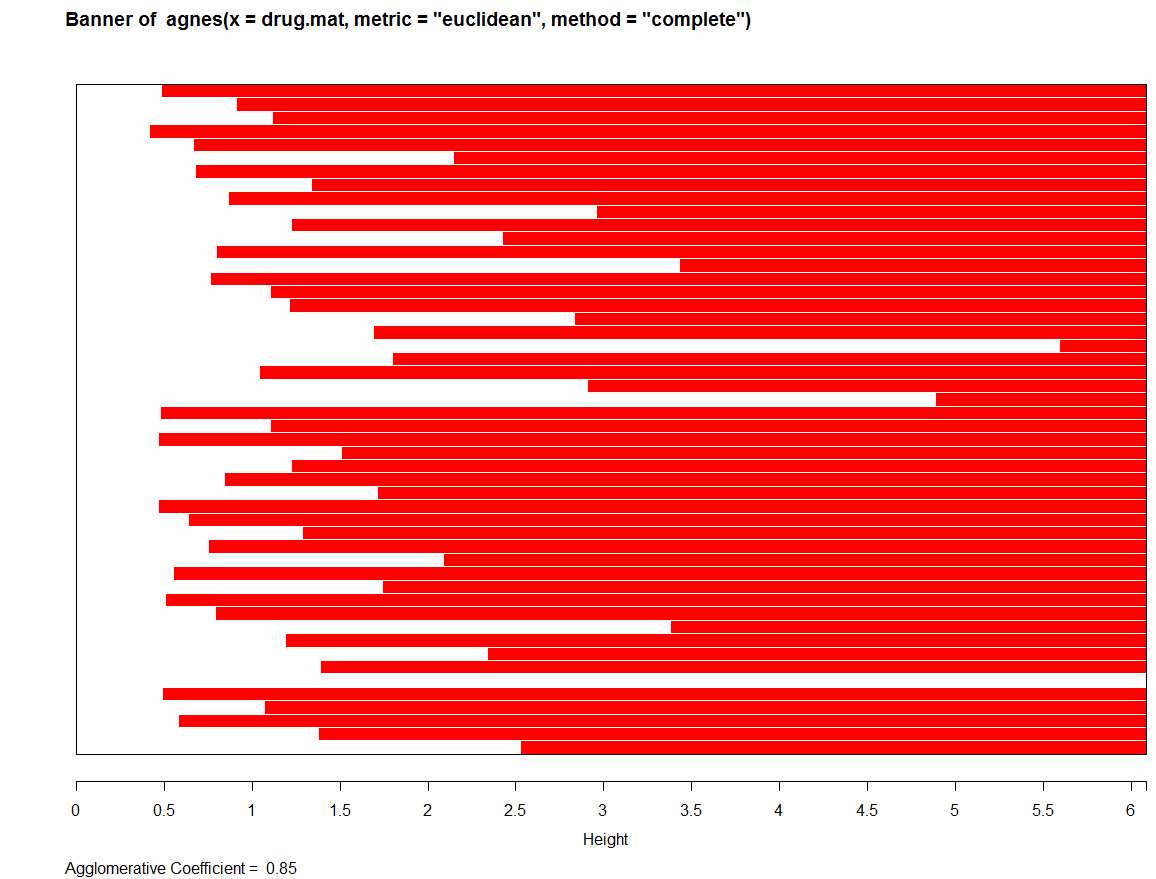
[1] "State" "DrugUse" "BingeDrink" "Poverty" "HSdrop" "Income"

> drug.mat = IllDrug[,2:5]

> drug.mat = scale(drug.mat)

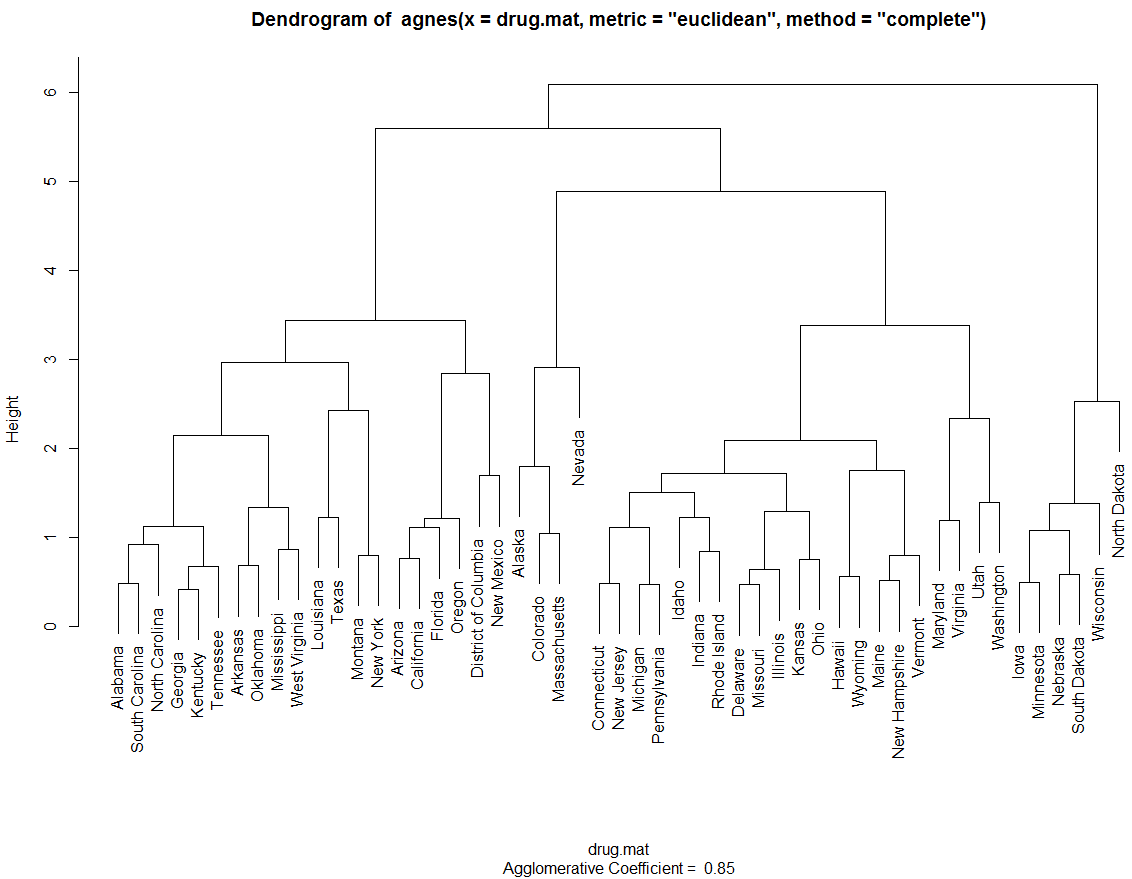
> drug.agnes = agnes(drug.mat,metric="euclidean",method="complete")

> plot(drug.agnes)



The *agglomerative coefficient* () measures the amount of clustering structure found. For each object *i* in the data set, denote by its dissimilarity to the first cluster it is merged with, divided by the dissimilarity of the merger in the last step of the algorithm. The is then defined as the average of all . **The closer the is to 1, the better the clustering structure found.**

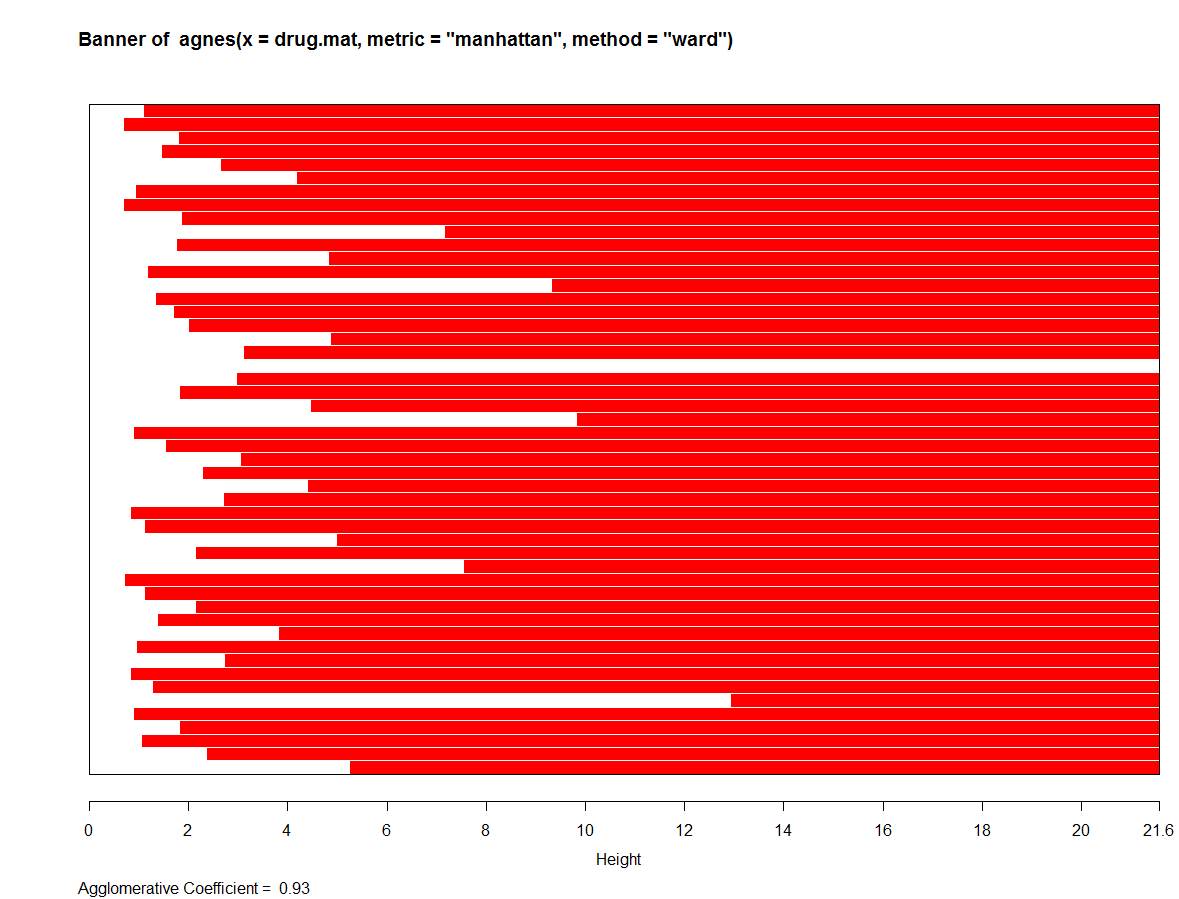
The banner plot is an alternative representation of the hierarchical cluster tree. The white bars are centered between observations that are joined together. The bar with the smallest height shows the two observations that were fused first. Subsequent bars show either pairs of observations that are merged to form new clusters or how observations are fused to existing clusters.

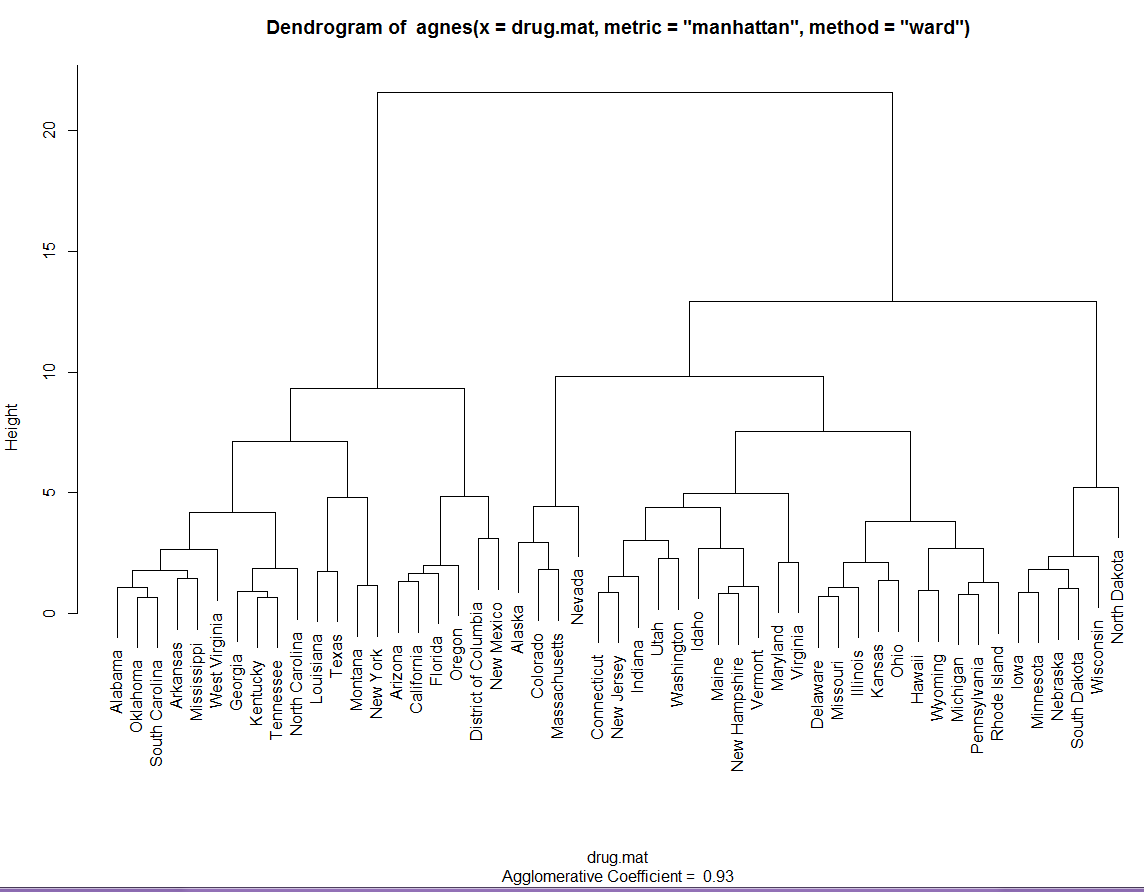


Let’s try a different metric and linkage.

> drug.agnes = agnes(drug.mat,metric="manhattan",method="ward")

> plot(drug.agnes)





This choice of metric & linkage appears to produce better clustering. As the result of agnes is a hierarchical cluster tree (dendrogram) we can use the cutree function to produce clusters.

> grps = cutree(drug.agnes,k=6)  
> clust.grps(drug.mat,grps,parcoord=T)

Cluster 1

=======================================================================

Alabama Arkansas Louisiana Mississippi New Mexico Oklahoma South Carolina Texas West Virginia

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

-0.8552758 -0.5697079 1.1572266 0.5653297 -1.1187964

Cluster 2

=======================================================================

Alaska Colorado Connecticut Massachusetts Nevada New Jersey

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

1.69899388 0.47359860 -0.89621240 0.04061277 1.43032125

Cluster 3

=======================================================================

Arizona California District of Columbia Florida Georgia Kentucky North Carolina Oregon Tennessee

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

0.2211054 -0.8846683 0.8171460 1.3017745 0.0892610

Cluster 4

=======================================================================

Delaware Hawaii Idaho Illinois Indiana Michigan Missouri Montana New Hampshire New York Pennsylvania Rhode Island Utah

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

0.19679152 0.17731370 -0.26687644 -0.23024314 0.09300278

Cluster 5

=======================================================================

Iowa Kansas Maine Minnesota Nebraska North Dakota Ohio South Dakota Vermont Wisconsin Wyoming

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

-0.5446319 1.0814484 -0.4845824 -1.2887538 -0.2502167

Cluster 6

=======================================================================

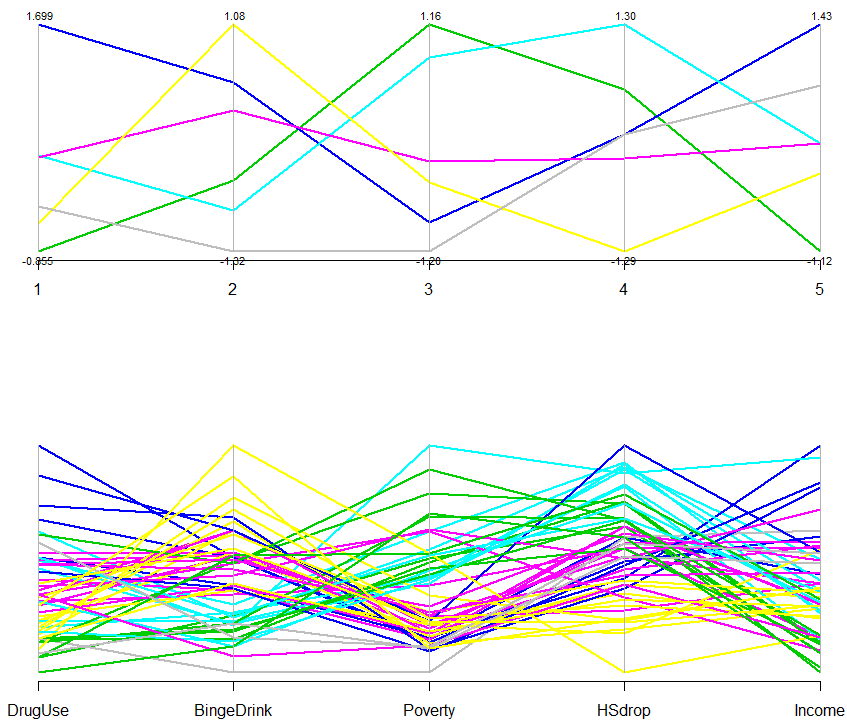
Maryland Virginia Washington

Variable means in this cluster are:

----------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop Income

-0.35125605 -1.31773889 -1.19742664 0.04061277 0.74241302



We can use boxplots to examine cluster differences on the individual variables.

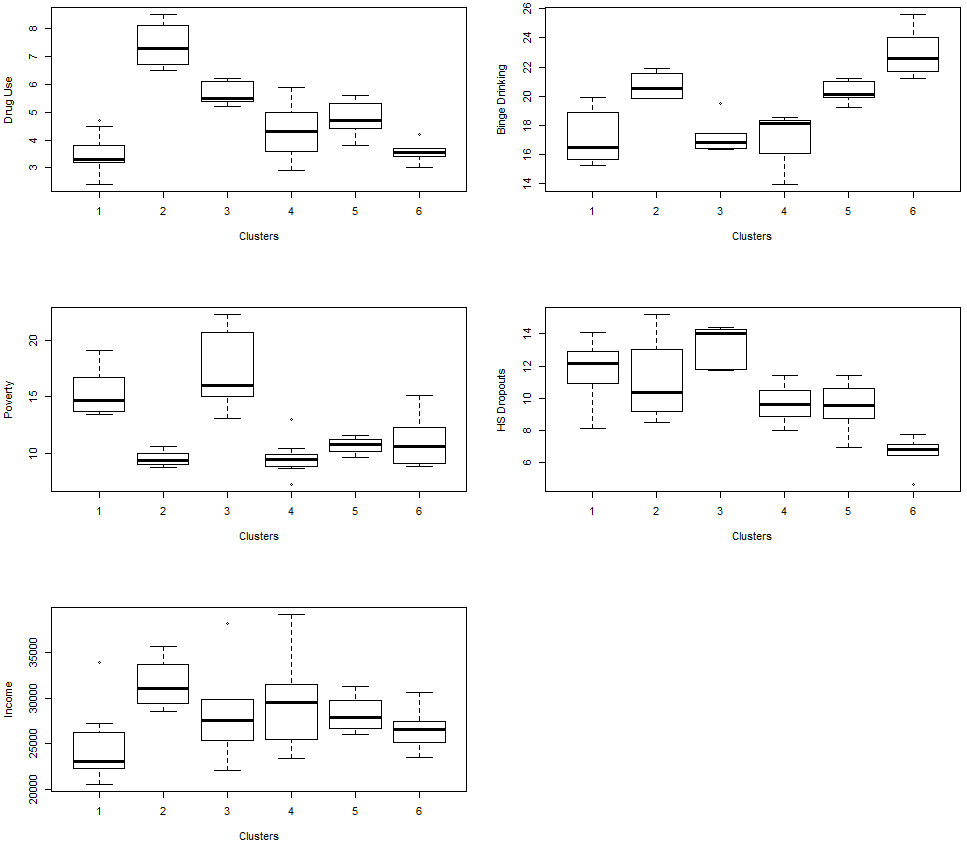
> par(mfrow=c(3,2))  
> boxplot(split(DrugUse,grps),xlab="Clusters",ylab="Drug Use")

> boxplot(split(BingeDrink,grps),xlab="Clusters",ylab="Binge Drinking")

> boxplot(split(Poverty,grps),xlab="Clusters",ylab="Poverty")

> boxplot(split(HSdrop,grps),xlab="Clusters",ylab="HS Dropouts")

> boxplot(split(Income,grps),xlab="Clusters",ylab="Income")



Divisive Hierarchical Clustering  
Divisive hierarchical clustering works in the opposite direction of agglomerative hierarchical clustering, as initially all objects to be clustered start in one large cluster and subsequent clusters are formed successively by splitting the objects into homogeneous groups. At the end of this process all objects are in their own group or cluster.

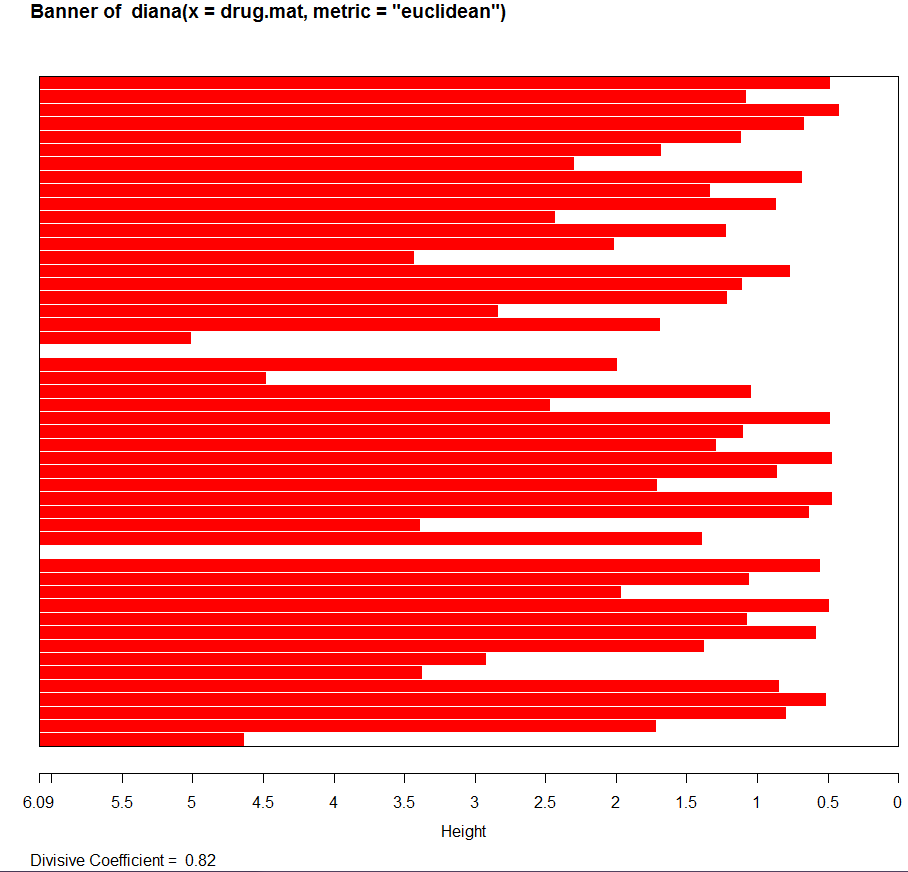
The function diana in the cluster library is a divisive hierarchical method. The initial cluster (at step 0) consists of one cluster containing all objects. In each subsequent step, the largest available cluster is split into two smaller clusters, until finally all clusters contain but a single object. The largest cluster is determined not by counting the number of observations in it but by computing the diameter of the clusters,

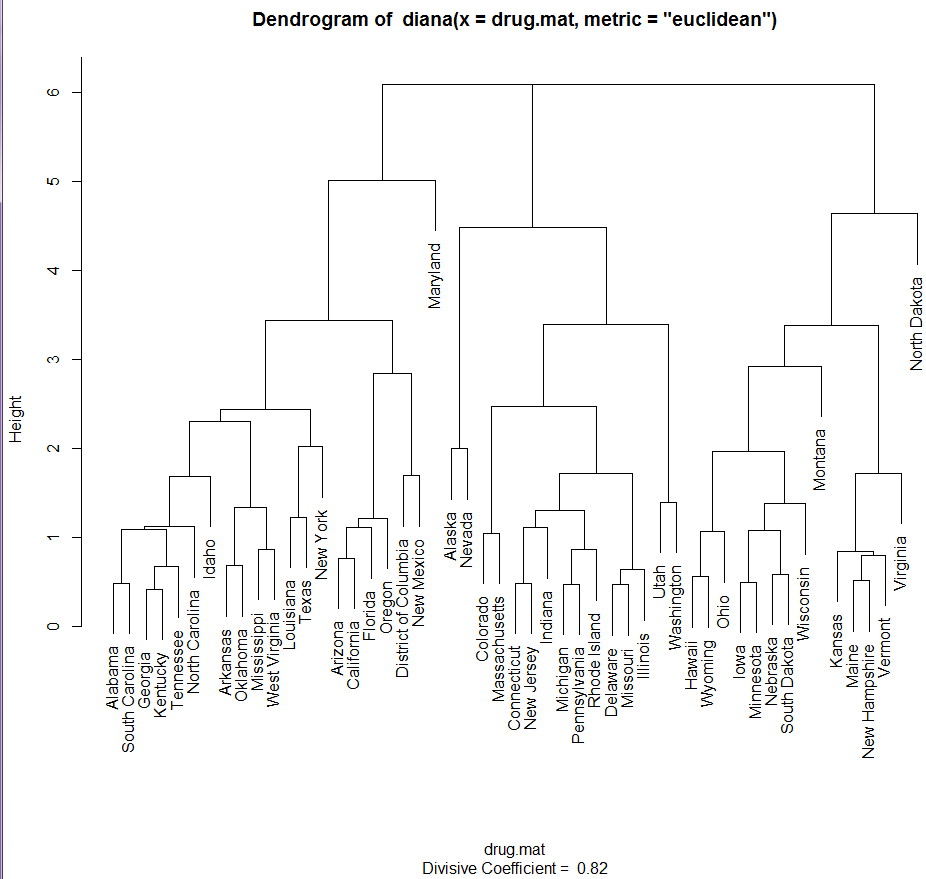
When a cluster is split into two clusters and it is done in such a way that the average dissimilarity of the two clusters is optimized.

Example 5.2: Illicit drug use in the U.S. (cont’d)

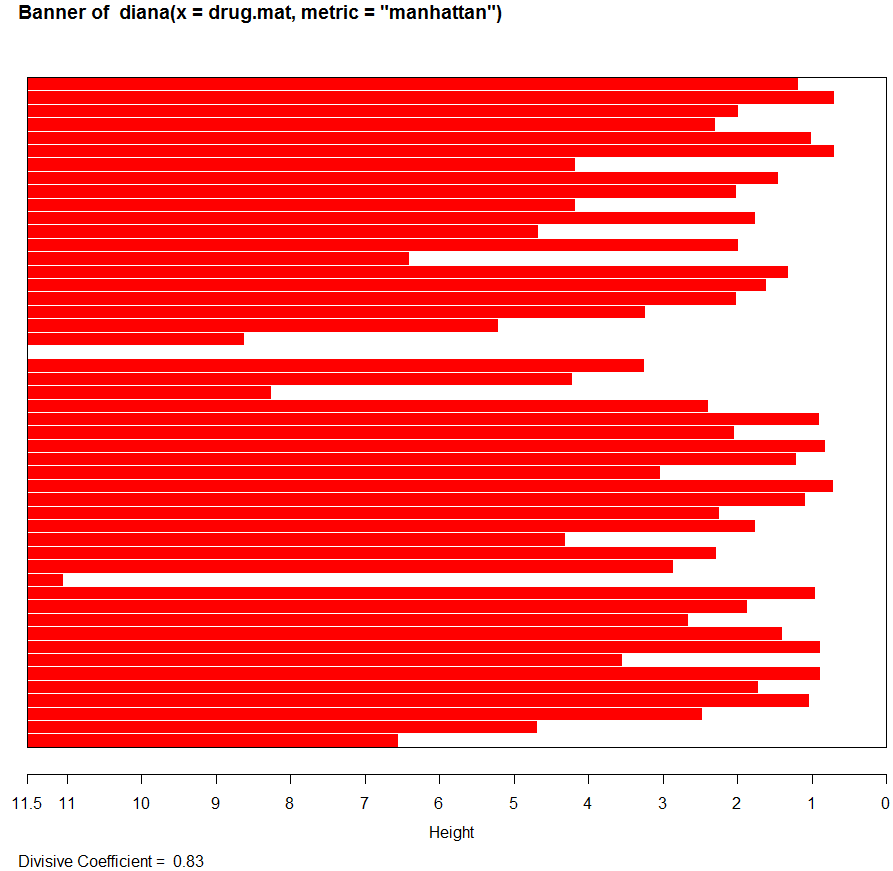
> drug.mat = Illdrug[,-1] 🡨 remove the first column containing the state names.

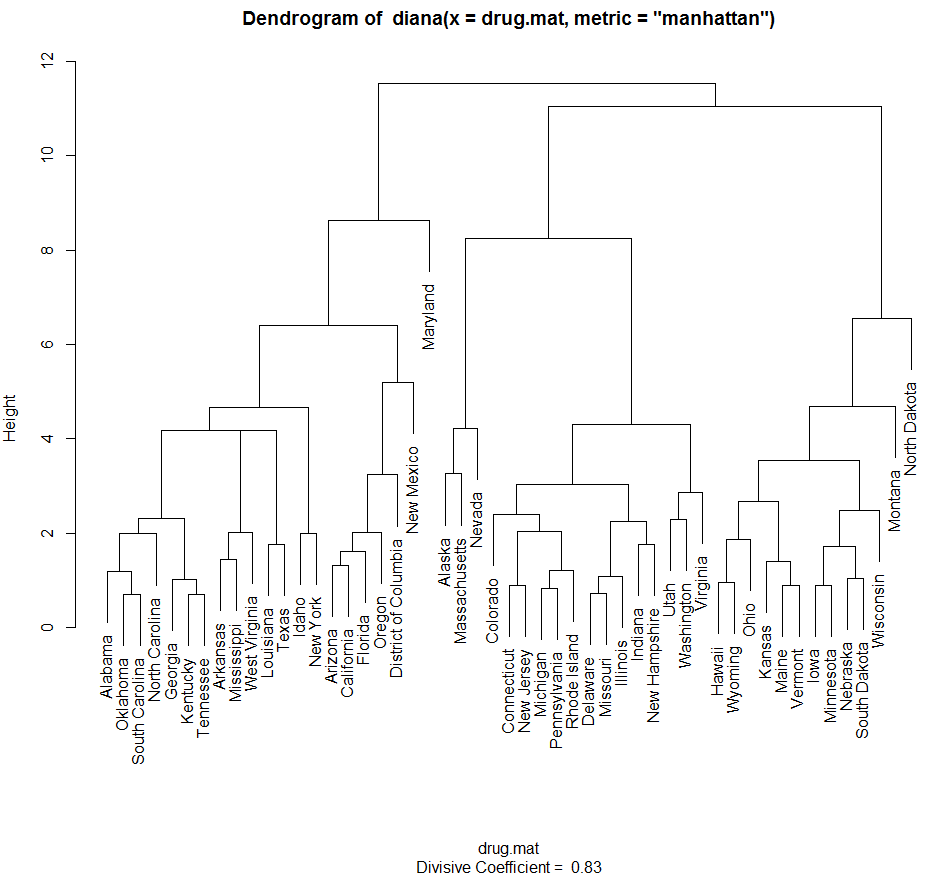
> drug.diana = diana(drug.mat,metric=”euclidean”,stand=T)





> drug.diana = diana(drug.mat,metric=”manhattan”,stand=T)  
> plot(drug.diana)





> grps = cutree(drug.diana,k=6)

> clust.grps(drug.mat,grps)

Cluster 1 consists of

=======================================================================

Alabama Arizona Arkansas California District of Columbia Florida Georgia Idaho Kentucky Louisiana Mississippi New Mexico New York North Carolina Oklahoma Oregon South Carolina Tennessee Texas West Virginia

Variable means in this cluster are:

-------------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop

-0.2846229 -0.6655083 0.9596560 0.8297594

Cluster 2 consists of

=======================================================================

Alaska Massachusetts Nevada

Variable means in this cluster are:

-------------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop

2.4678376 0.9001075 -0.8281963 0.4686713

Cluster 3 consists of

=======================================================================

Colorado Connecticut Delaware Illinois Indiana Michigan Missouri New Hampshire New Jersey Pennsylvania Rhode Island Utah Virginia Washington

Variable means in this cluster are:

-------------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop

0.32056691 -0.01477756 -0.76018016 -0.13396411

Cluster 4 consists of

=======================================================================

Hawaii Iowa Kansas Maine Minnesota Montana Nebraska Ohio South Dakota Vermont Wisconsin Wyoming

Variable means in this cluster are:

-------------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop

-0.3448490 0.8312099 -0.4443910 -1.1607128

Cluster 5 consists of

=======================================================================

Maryland

Variable means in this cluster are:

-------------------------------------------------------------------------

DrugUse BingeDrink Poverty HSdrop

-0.9150748 -1.8820430 -1.5180741 0.2063128

Cluster 6 consists of

=======================================================================

North Dakota

Variable means in this cluster are:

-------------------------------------------------------------------------

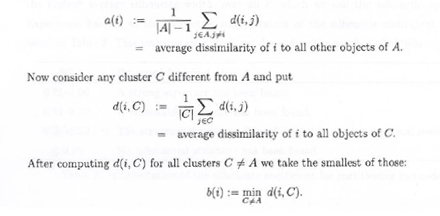
DrugUse BingeDrink Poverty HSdrop

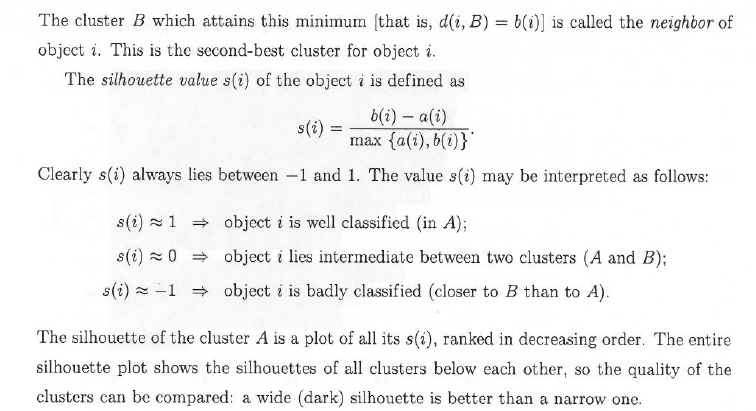
-1.1457279 2.7242534 0.7847574 -2.4034634

For this particular data set divisive clustering does not produce very satisfactory results as two of the six clusters formed contain a single observation. This seems to be the case with other data sets on which diana is used, it tends to place outliers on branches high on tree so when clusters are formed by cutting the tree they end up as singleton clusters.

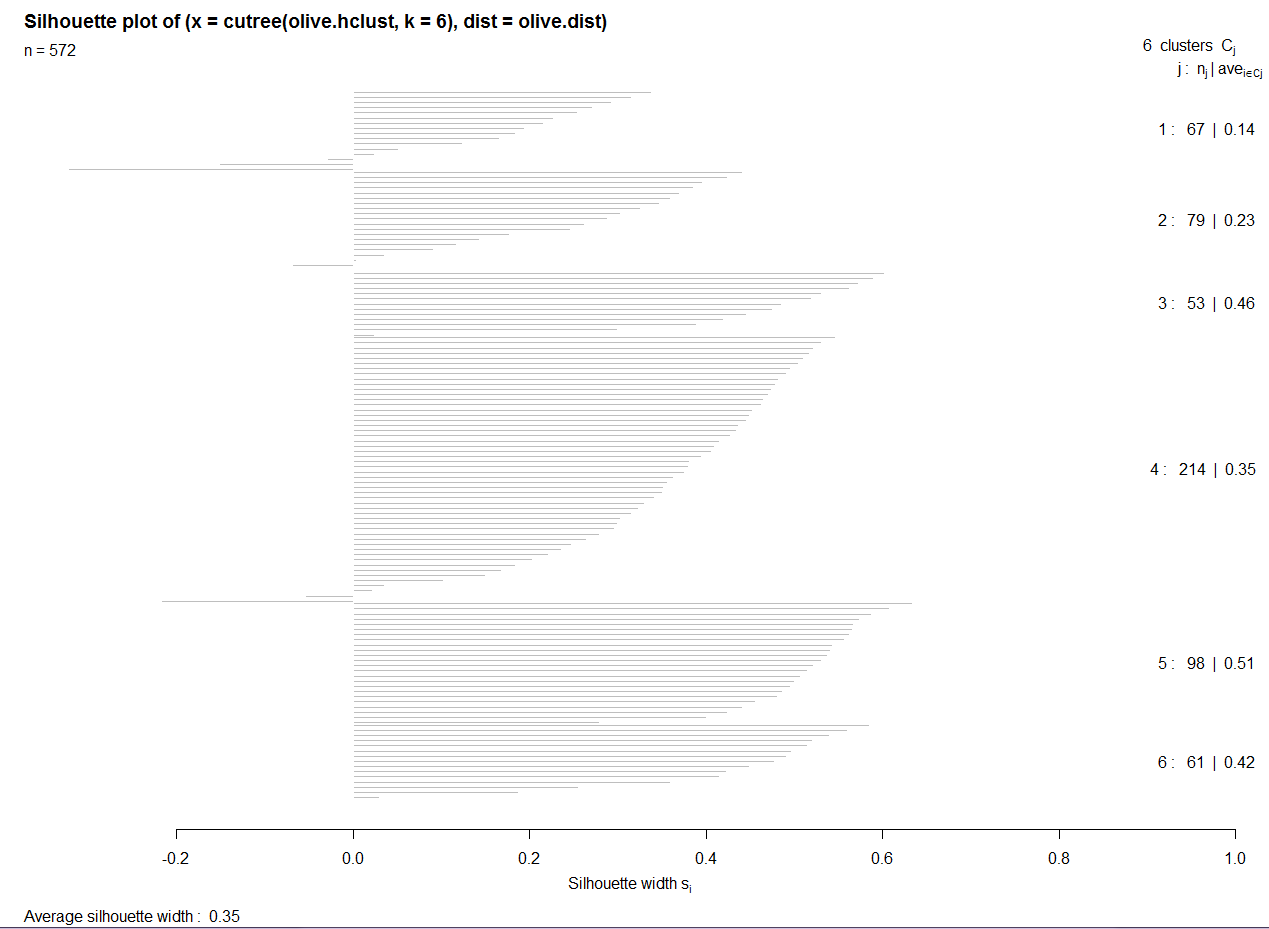
Silhouette Plots

*Silhouette Plots* are a graphical method for determining the goodness of a clustering. For each object in an arbitrary cluster denoted by *A* we define the following:

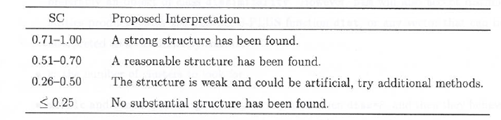




A sample silhouette plot is shown below:



In general, we can run a cluster analysis with different choices for the number of clusters *k* to compare the resulting silhouette plots. We can select the value of *k* yielding the highest average silhouette width, over all *k* clusters, which is called the *silhouette coefficient*. A subjective interpretation of the silhouette coefficient () is listed in the table below:



It is my experience that getting values greater than .50 is rare, especially for large datasets thus in my opinion the “Proposed Interpretation” needs to be taken with a *grain of salt*.

Example 5.2: Illicit drug use in U.S. (cont’d)

First we perform hierarchical cluster analysis using the taxi-cab metric and Ward’s linkage.

> drug.mat = scale(IllDrug[,-1])

> drug.dist = dist(drug.mat,method="manhattan")

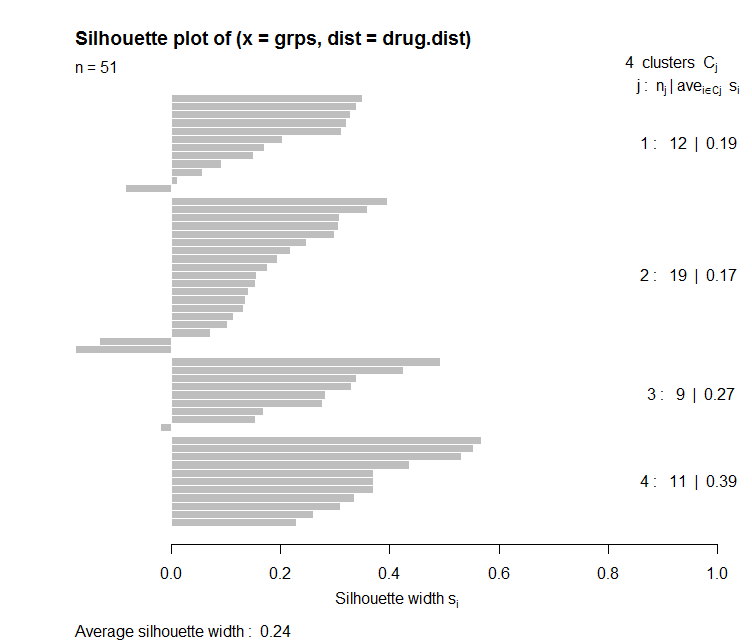
> drug.hc = hclust(drug.dist,method="ward")

> plot(drug.hc)

> grps = cutree(drug.hc,k=4)

> si = silhouette(grps,drug.dist) 🡨 give cluster memberships and pair-wise distances

> plot(si)



At this point we can try different combinations of metrics, linkages, and the number of clusters to maximize the average silhouette width. I tried several and got a maximum of around 0.29 for these data.

Example 5.4: Mushrooms  
  
> library(cba)

> names(Mushrooms.train)

[1] "Poisonous" "x1" "x2" "x3" "x4" "x5" "x6" "x7" "x8"

[10] "x9" "x10" "x11" "x12" "x13" "x14" "x15" "x16" "x17"

[19] "x18" "x19" "x20" "x21" "x22"   
  
> mush.subset = Mushrooms.train[,-c(1,12,17)] 🡨 remove mushrooms with missing   
 values & edibility.

> mush.dummy = as.dummy(mush.subset) 🡨 requires library cba

> mush.dist = dist(mush.dummy,method="jaccard")

> mush.hc = hclust(mush.dist,method="ward.D")

> mush.hc2 = hclust(mush.dist,method="ward.D2")

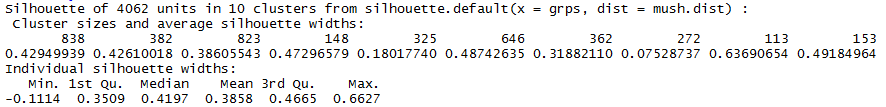
> grps = cutree(mush.hc,k=10) 🡨 we can try various choices for *k* trying

maximize the mean silhouette width.

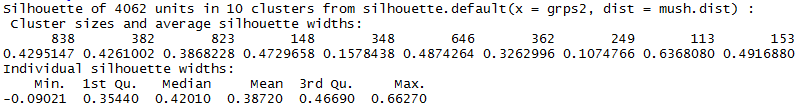
> grps2 = cutree(mush.hc2,k=10)

> si1 = silhouette(grps,mush.dist) 🡨 silhouette takes groups/clusters & distance   
 matrix used in clustering as arguments.

> si2 = silhouette(grps2,mush.dist)

> summary(si1)  


> summary(si2)

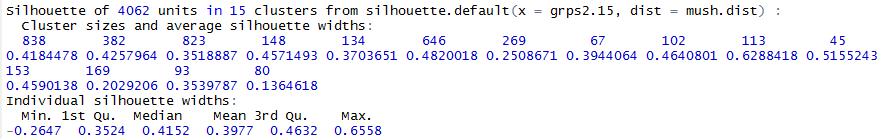


The maximum was obtained using using Ward D2 clustering, though most choices between 10 – 15 produced very similar mean silhouette widths.

> grps2.15 = cutree(mush.hc2,k=15)

> si2.15 = silhouette(grps2.15,mush.dist)

> summary(si2.15)



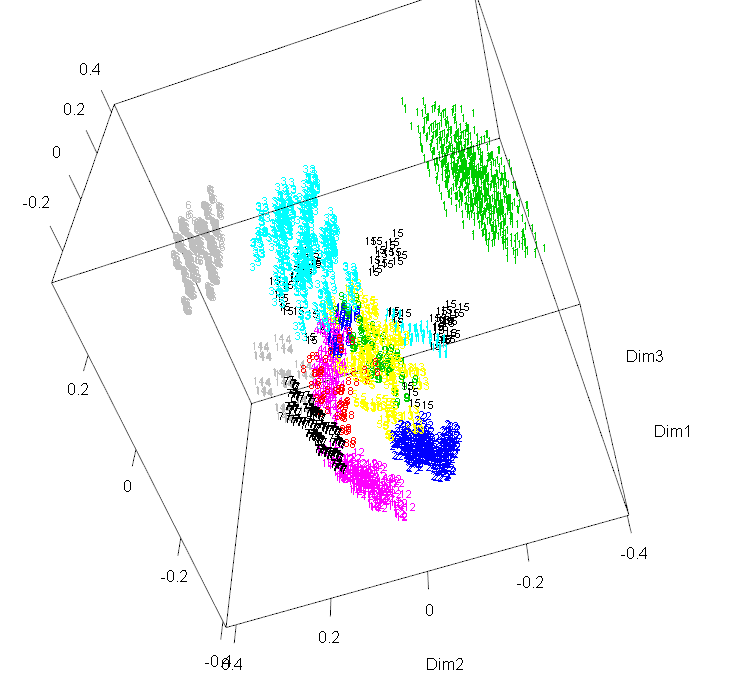
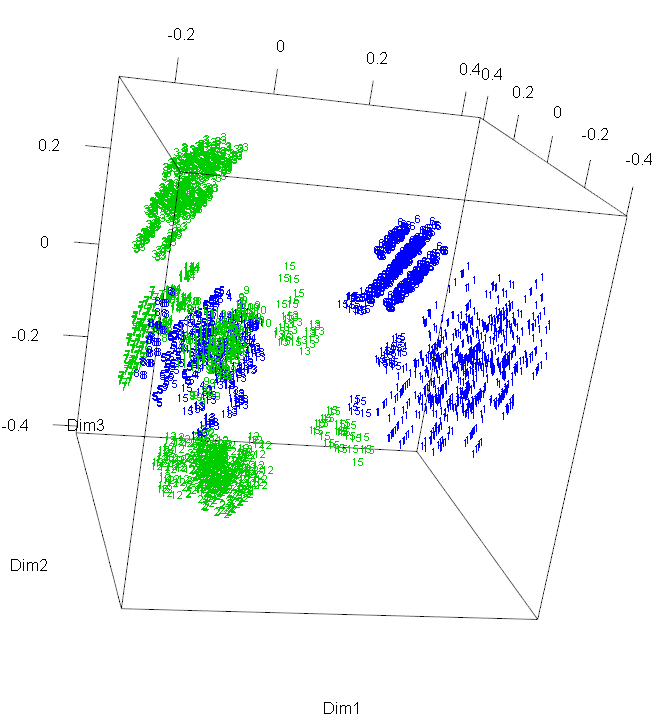
> mush.mds = cmdscale(mush.dist,k=3)

> library(rgl)

> plot3d(mush.mds,type="n",xlab="Dim1",ylab="Dim2",zlab="Dim3")

> text3d(mush.mds,texts=grps2.15,col=as.numeric(Mushrooms.train[,1])

,cex=.6)



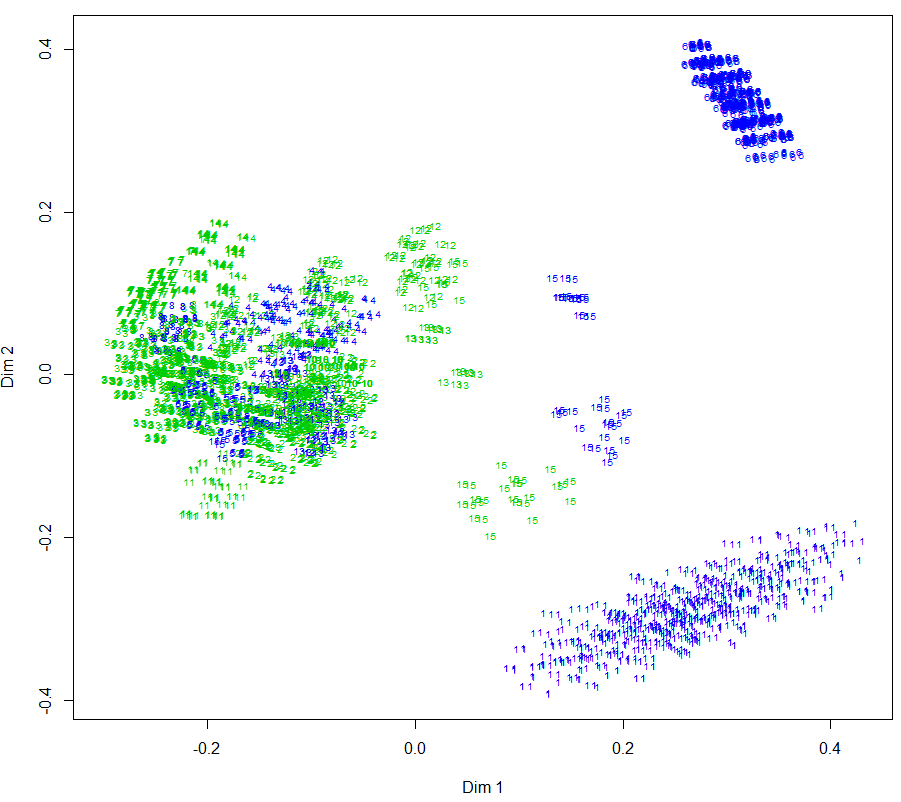
We would also use cluster number here, i.e. grps2.15.

We can also create some 2-D plots for the pairs of dimensions returned my MDS.

> plot(mush.mds[,1:2],type="n",xlab="Dim 1",ylab="Dim 2")

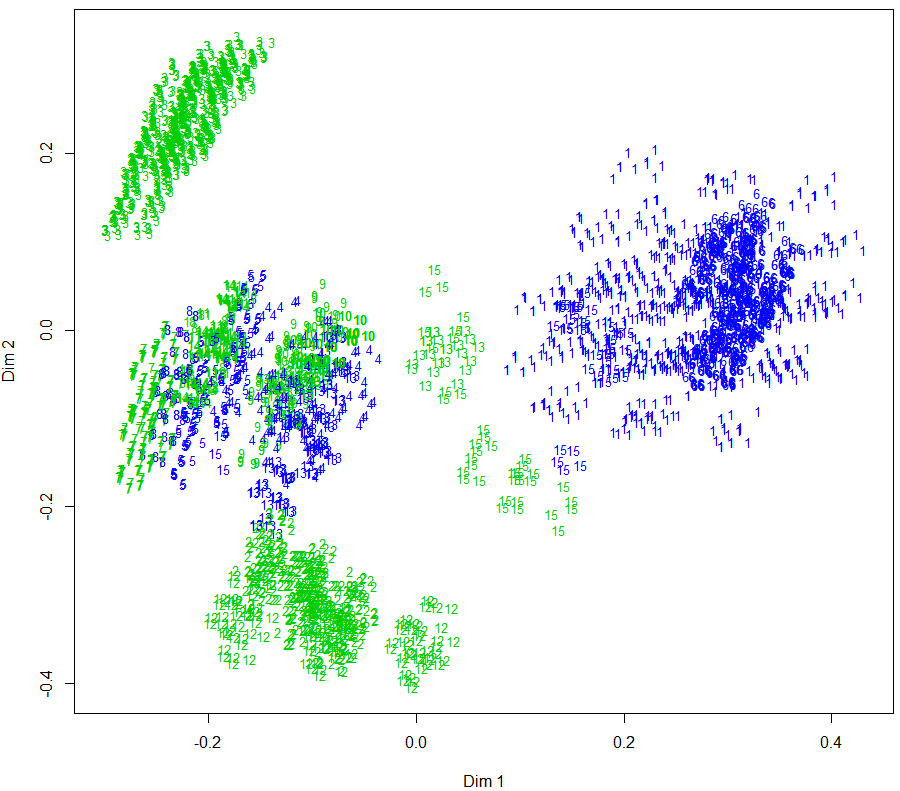
> text(mush.mds[,1:2],labels=grps2.15,

col=as.numeric(Mushrooms.train[,1])+2,cex=.6)



> plot(mush.mds[,c(1,3)],type="n",xlab="Dim 1",ylab="Dim 2")

> text(mush.mds[,c(1,3)],labels=grps2.15,  
col=as.numeric(Mushrooms.train[,1])+2,cex=.8)



> plot(mush.mds[,c(2,3)],type="n",xlab="Dim 2",ylab="Dim 3")

> text(mush.mds[,c(2,3)],labels=grps2.15,  
col=as.numeric(Mushrooms.train[,1])+2,cex=.8)

