#Lab 1- Principal component analysis

In [1]:
```
data(USArrests)
```

In [2]:
```
states=row.names(USArrests)  #row of states
states
```

```
'Alabama'  'Alaska'  'Arizona'  'Arkansas'  'California'  'Colorado'  'Connecticut'
'Delaware'  'Florida'  'Georgia'  'Hawaii'  'Idaho'  'Illinois'  'Indiana'  'Iowa'  'Kansas'
'Kentucky'  'Louisiana'  'Maine'  'Maryland'  'Massachusetts'  'Michigan'  'Minnesota'
'Mississippi'  'Missouri'  'Montana'  'Nebraska'  'Nevada'  'New Hampshire'
'New Jersey'  'New Mexico'  'New York'  'North Carolina'  'North Dakota'  'Ohio'
'Oklahoma'  'Oregon'  'Pennsylvania'  'Rhode Island'  'South Carolina'  'South Dakota'
'Tennessee'  'Texas'  'Utah'  'Vermont'  'Virginia'  'Washington'  'West Virginia'
'Wisconsin'  'Wyoming'
```

In [3]:
```
names(USArrests) #columns of four variables
```

```
'Murder'  'Assault'  'UrbanPop'  'Rape'
```

In [4]:
```
apply(USArrests,2,mean)  #mean- 2 means col
```

```
  Murder   7.788
 Assault   170.76
UrbanPop   65.54
    Rape   21.232
```

In [5]:
```
#on avg. three times as many rapes as murders, and more than eight
#times as many assaults as rapes
```

In [6]:
```
apply(USArrests,2,var) #computing variance of four variables
```

```
  Murder   18.9704653061224
 Assault   6945.16571428571
UrbanPop   209.518775510204
    Rape   87.7291591836735
```

In [7]:

```r
pr.out=prcomp(USArrests,scale=TRUE)
#prcomp()centers the variables to have mean 0, using scale=TRUE
#we scale the variables to have std deviation 1
```

In [8]:

```r
names(pr.out)
```

```
'sdev'   'rotation'   'center'   'scale'   'x'
```

In [9]:

```r
pr.out$center  #correspond to mean
```

```
 Murder   7.788
Assault   170.76
UrbanPop  65.54
   Rape   21.232
```

In [10]:

```r
pr.out$scale  #correspond to sd
```

```
  Murder  4.35550976420929
 Assault  83.3376608400171
UrbanPop  14.4747634008368
    Rape  9.36638453105965
```

In [11]:

```r
pr.out$rotation
#rotation gives principal component loadings
```

```
                PC1         PC2         PC3          PC4
  Murder  -0.5358995   0.4181809  -0.3412327   0.64922780
 Assault  -0.5831836   0.1879856  -0.2681484  -0.74340748
UrbanPop  -0.2781909  -0.8728062  -0.3780158   0.13387773
    Rape  -0.5434321  -0.1673186   0.8177779   0.08902432
```

In [12]:

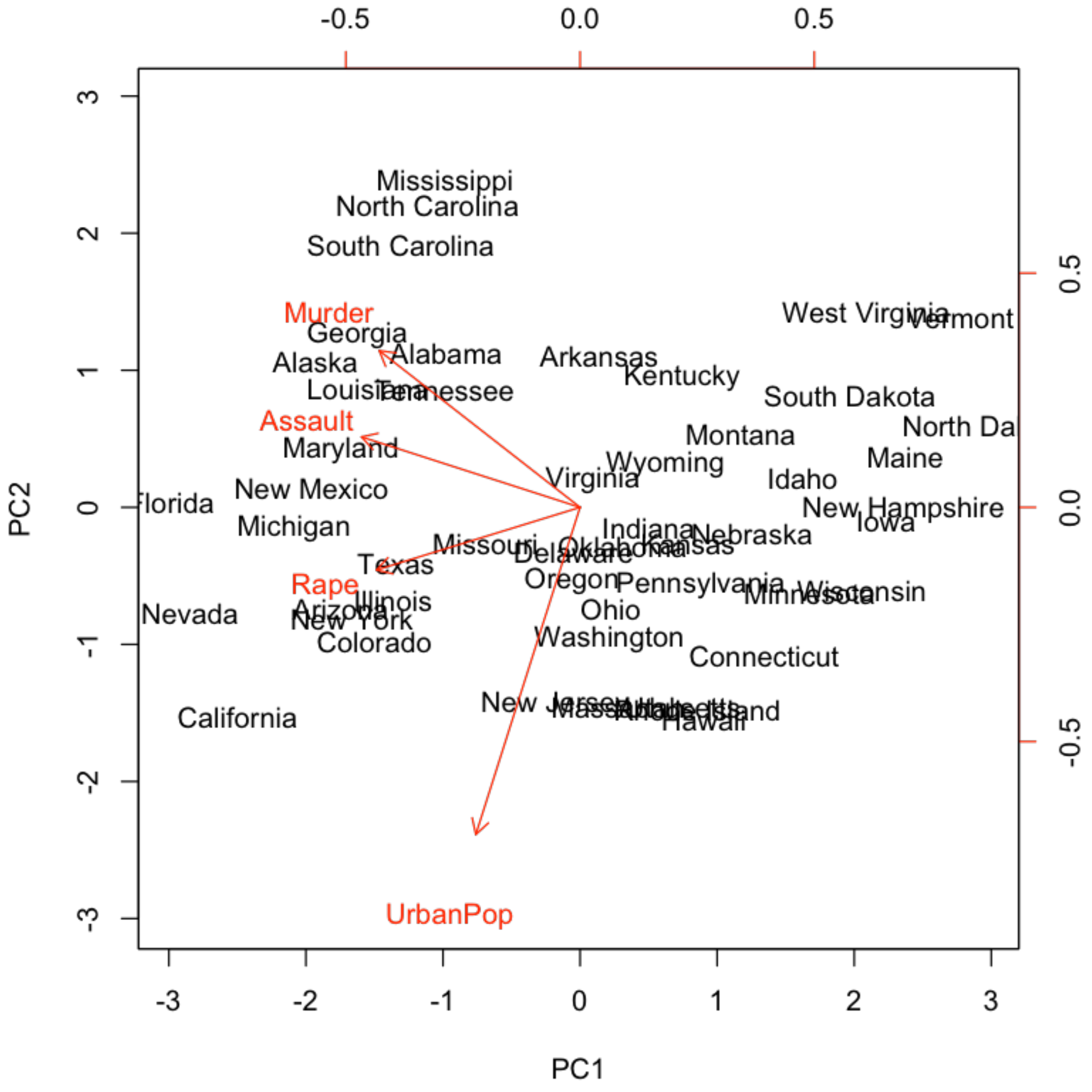
```r
#4 distict pca which is expected,
#in general min(n − 1, p)informative principal components
```

In [13]:

```r
dim(pr.out$x)  #matrix x
```

```
50   4
```

In [14]:
```
biplot(pr.out,scale=0)
#with scale=0 arrows are scaled to represent the loadings
```

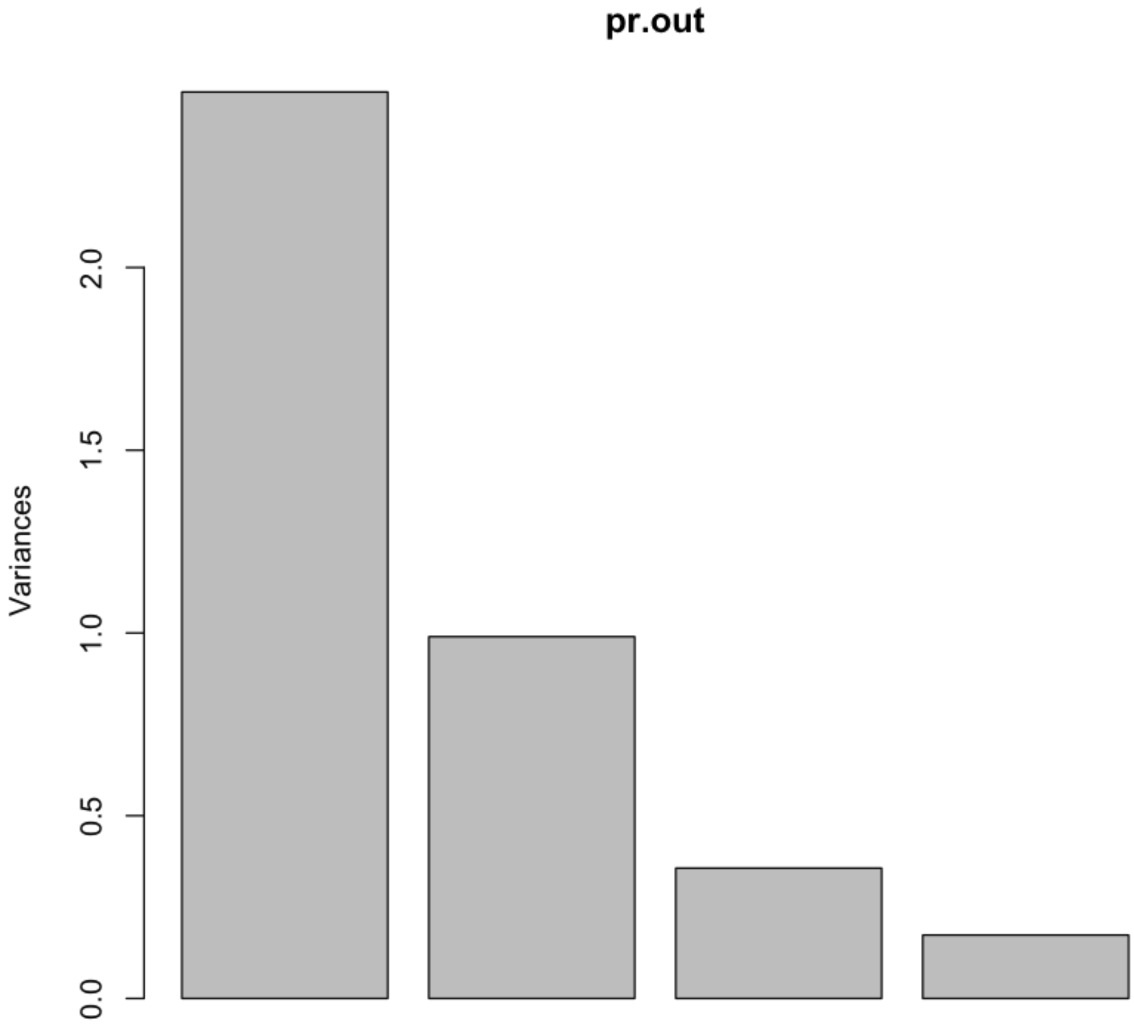


In [15]:
```
pr.out$rotation=-pr.out$rotation #making mirror image
```

In [16]:
```
pr.out$x=-pr.out$x
```

In [17]:
```
plot(pr.out,scale=0)
```

```
Warning message in plot.window(xlim, ylim, log = log, ...):
“"scale" is not a graphical parameter”Warning message in title(main =
main, sub = sub, xlab = xlab, ylab = ylab, ...):
“"scale" is not a graphical parameter”Warning message in axis(if (hor
iz) 1 else 2, cex.axis = cex.axis, ...):
“"scale" is not a graphical parameter”
```



In [18]:
```
pr.out$sdev   #sd of each pc
```

```
1.57487827439123   0.994869414817764   0.597129115502527   0.41644938195396
```

In [19]:

```
pr.var=pr.out$sdev^2  #calculating variance
pr.var
```

```
2.48024157914949   0.98976515253984   0.35656318058083   0.173430087729835
```

In [20]:

```
#To compute the proportion of variance explained by each PC
#we simply divide the variance explained by each PC
#by the total variance explained by all four PC

pve=pr.var/sum(pr.var)
pve
```

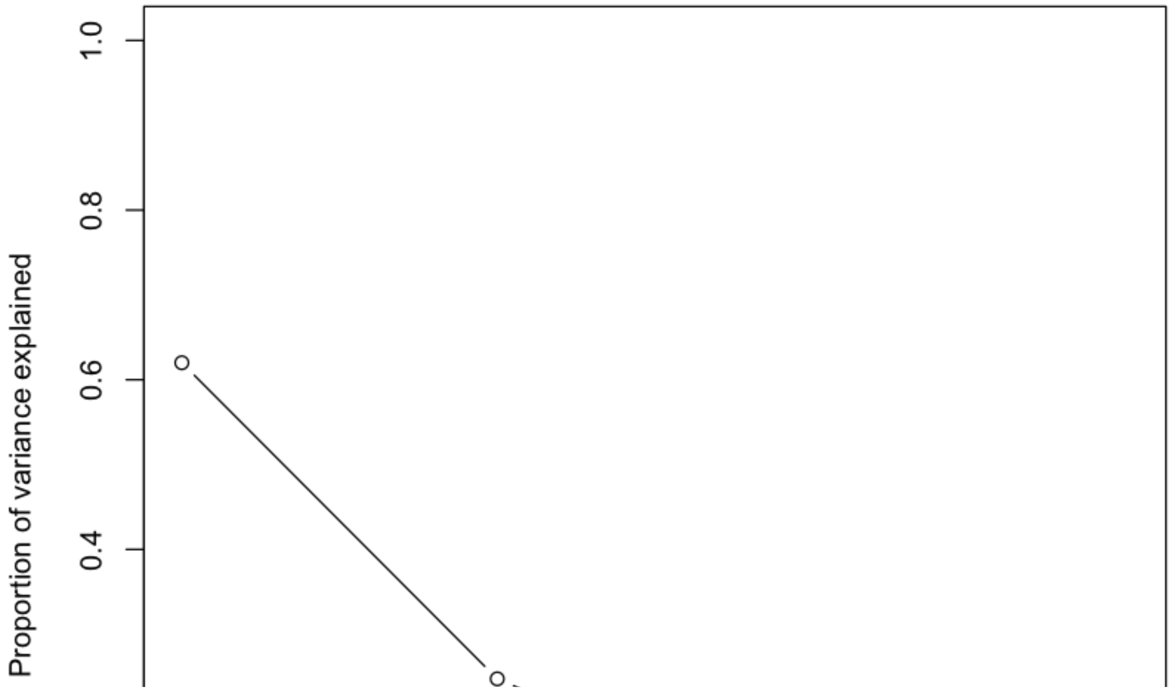
```
0.620060394787374   0.24744128813496   0.0891407951452075   0.0433575219324588
```

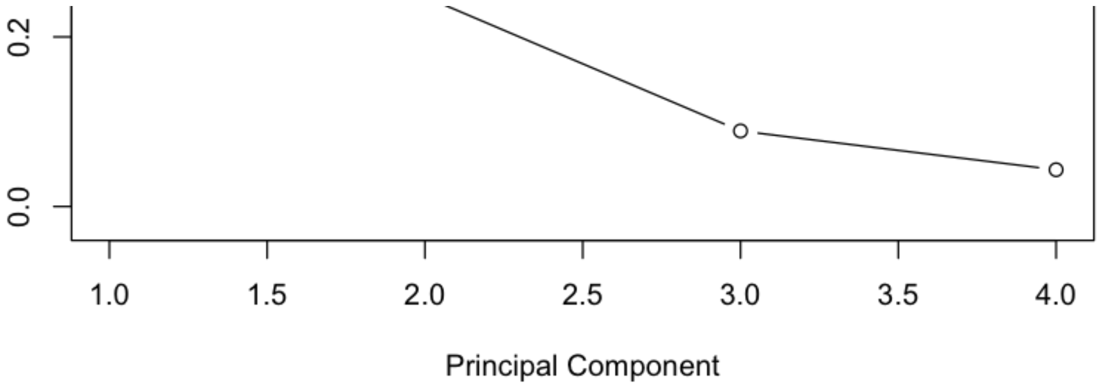
In [21]:

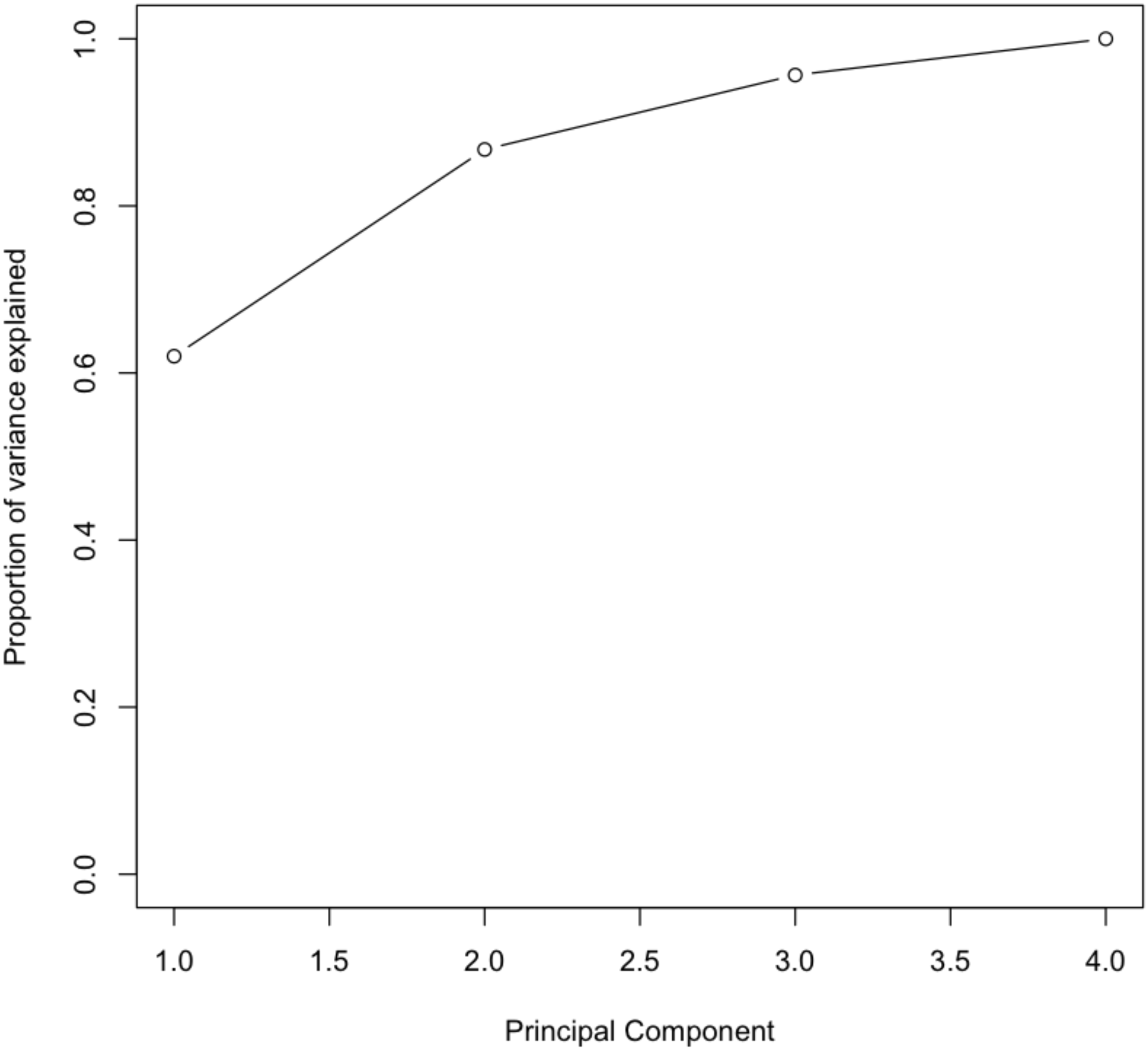
```
#PC1 explains 62% of the variance in the data, PC2 24.7% and so or
```

In [22]:

```
plot(pve, xlab="Principal Component", ylab="Proportion of variance
    ylim=c(0,1), type="b")

plot(cumsum(pve),xlab="Principal Component",
     ylab="Proportion of variance explained",
    ylim=c(0,1), type="b") #commulative pve

#cumsum()computes the cumulative sum of the elements of a numeric
```




0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component


Proportion of variance explained
1.0
0.8
0.6
0.4
0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component

#Lab 2: Clustering

In [23]:

```
#k-means clustering

```

In [24]:

```
set.seed(2)
x=matrix(rnorm(50*2),ncol=2)
x[1:25,1]=x[1:25,1]+3
x[1:25,2]=x[1:25,2]-4
```

In [25]:

```
x
```

```
2.10308545   -4.838287148
3.18484918   -1.933698644
4.58784533   -4.562247053
1.86962433   -2.724284488
2.91974824   -5.047572627
3.13242028   -5.965878241
3.70795473   -4.322971094
2.76030198   -3.064137473
4.98447394   -2.860770197
2.86121299   -2.328381233
3.41765075   -5.788242207
3.98175278   -1.968757481
2.60730464   -4.703144333
1.96033102   -3.841835237
4.78222896   -3.493765203
0.68893092   -4.819995106
3.87860458   -5.998846995
3.03580672   -4.479292591
4.01282869   -3.915820096
3.43226515   -4.895486611
5.09081921   -4.921275666
1.80007418   -3.669550497
4.58963820   -4.141660809
```

```
 4.95465164  -3.565152238
 3.00493778  -4.053722626
-2.45170639  -0.907110376
 0.47723730   1.303512232
-0.59655817   0.771789776
 0.79220327   1.052525595
 0.28963671  -1.410038341
 0.73893860   0.995984590
 0.31896040  -1.695764903
 1.07616435  -0.533372143
-0.28415772  -1.372269451
-0.77667527  -2.207919779
-0.59566050   1.822122519
-1.72597978  -0.653393411
-0.90258448  -0.284681219
-0.55906191  -0.386949604
-0.24651257   0.386694975
-0.38358623   1.600390852
-1.95910318   1.681154956
-0.84170506  -1.183606388
 1.90354747  -1.358457254
 0.62249393  -1.512670795
 1.99092044  -1.253104899
-0.30548372   1.959357077
-0.09084424   0.007645872
-0.18416145  -0.842615198
-1.19876777  -0.601160105
```

In [26]:

```
km.out=kmeans(x,2,nstart=20)
```

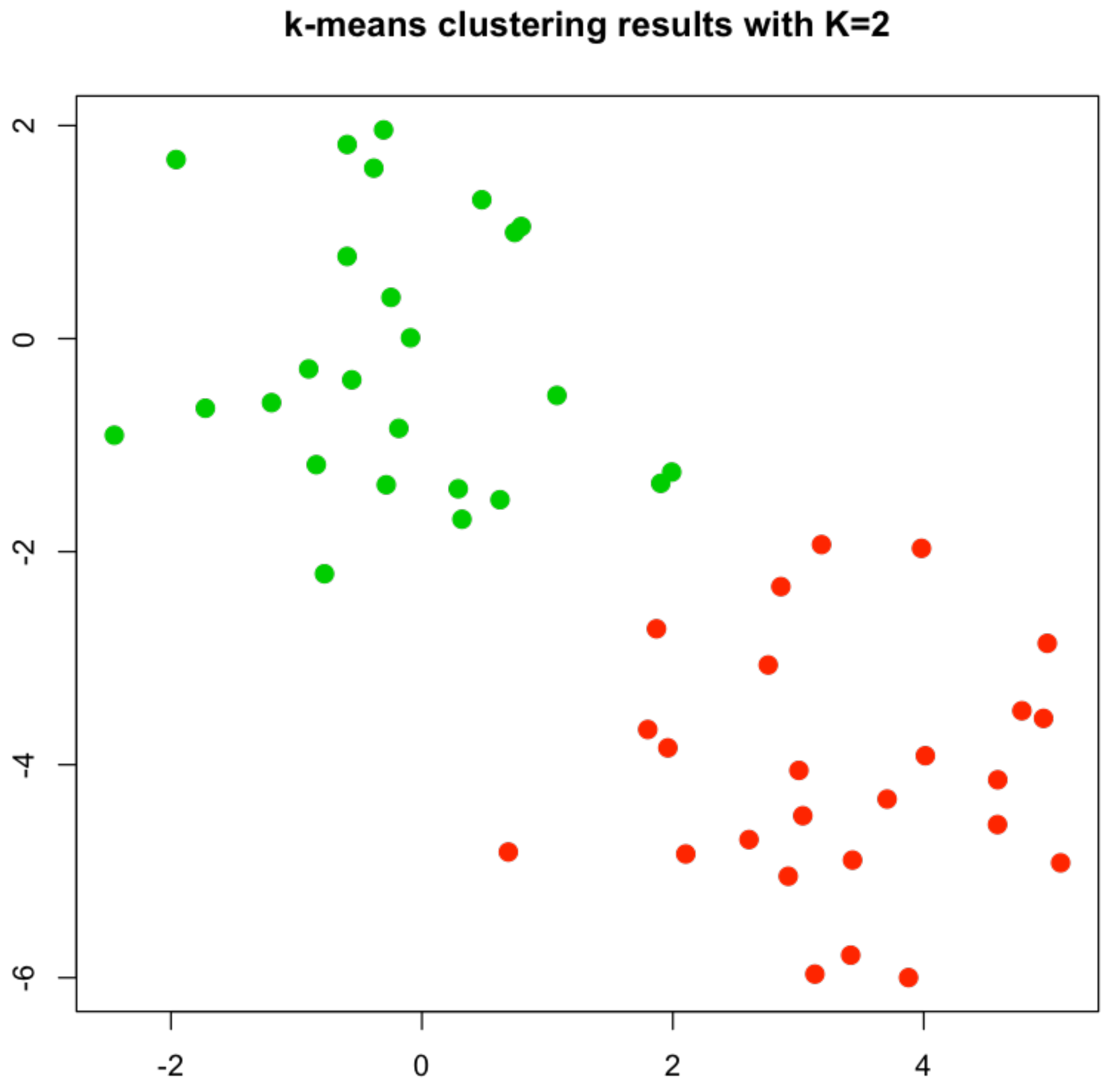
In [27]:

```
km.out
```

```
K-means clustering with 2 clusters of sizes 25, 25

Cluster means:
        [,1]       [,2]
1  3.3339737 -4.0761910
2 -0.1956978 -0.1848774

Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 63.20595 65.40068
 (between_SS / total_SS =  72.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.
withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

In [28]:

```
plot(x,col=(km.out$cluster+1),main="k-means clustering results wit
    xlab="",ylab="",pch=20,cex=2)
```



In [29]:

```
set.seed (4)
km.out=kmeans(x,3,nstart=20)
km.out
```

```
K-means clustering with 3 clusters of sizes 17, 23, 10

Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2 2 2
2 2 2 2 2 2
[39] 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.
withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

In [30]:

```
km.out=kmeans(x,2,nstart=20)
```

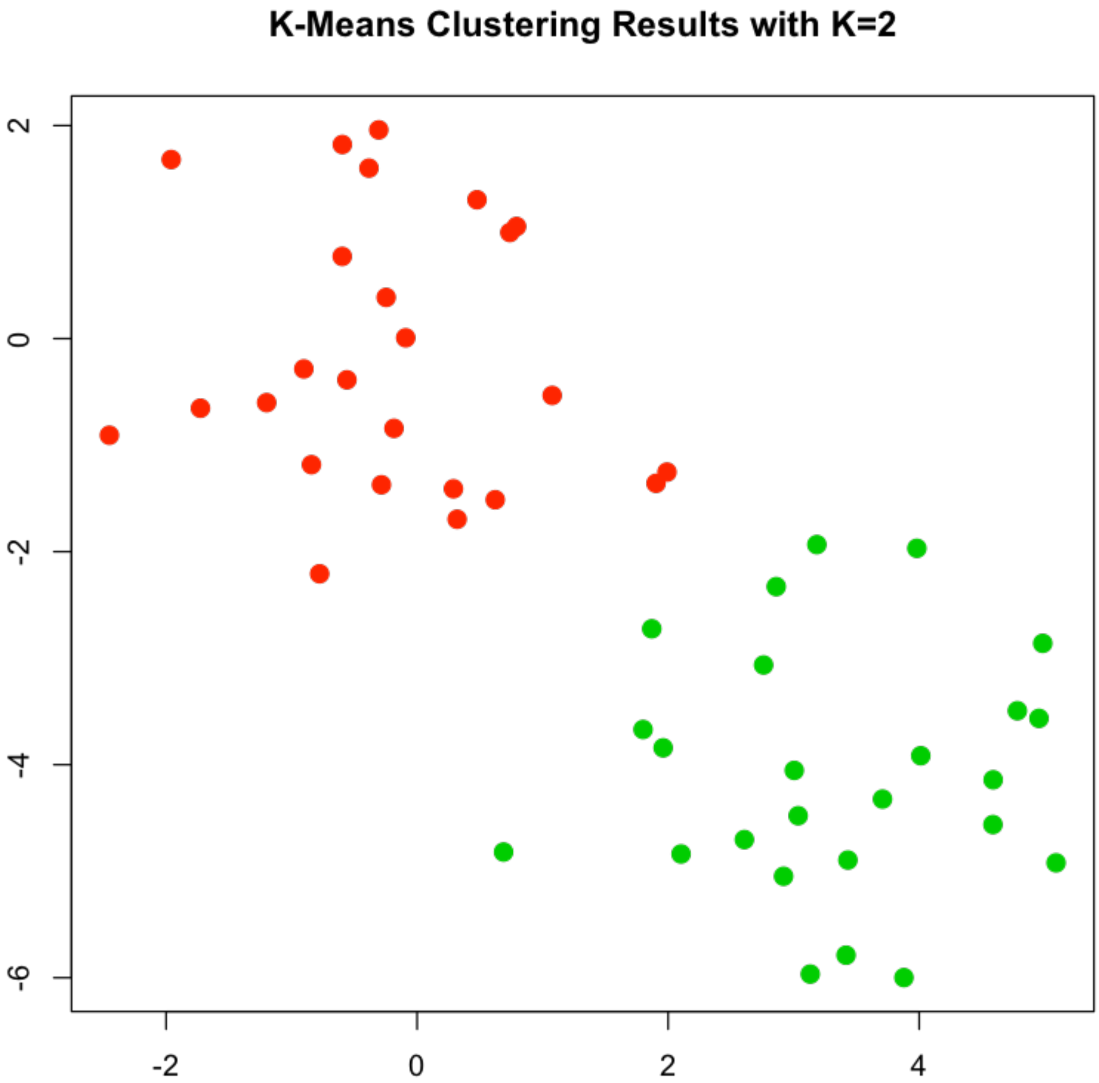
In [31]:

```
km.out$cluster
```

```
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

In [32]:

```
plot(x, col=(km.out$cluster +1), main="K-Means Clustering Results
     xlab="", ylab="", pch=20, cex=2)
```



In [33]:

```
set.seed(6)
km.out=kmeans(x,3,nstart=20)
km.out
```

```
K-means clustering with 3 clusters of sizes 10, 23, 17

Cluster means:
        [,1]        [,2]
1  2.3001545 -2.69622023
2 -0.3820397 -0.08740753
3  3.7789567 -4.56200798

Clustering vector:
 [1] 3 1 3 1 3 3 3 1 3 1 3 1 3 1 3 1 3 3 3 3 3 1 3 3 3 2 2 2 2 2 2 2
2 2 2 2 2 2
[39] 2 2 2 2 2 1 2 1 2 2 2 2

Within cluster sum of squares by cluster:
[1] 19.56137 52.67700 25.74089
 (between_SS / total_SS =  79.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.
withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

In [34]:

```
set.seed(4)
km.out=kmeans(x,3,nstart=1)
km.out$tot.withinss
```

104.331921973392

In [35]:

```
km.out=kmeans(x,3,nstart=20)
km.out$tot.withinss
```

97.9792674793981

#Hierarchical Clustering

In [36]:

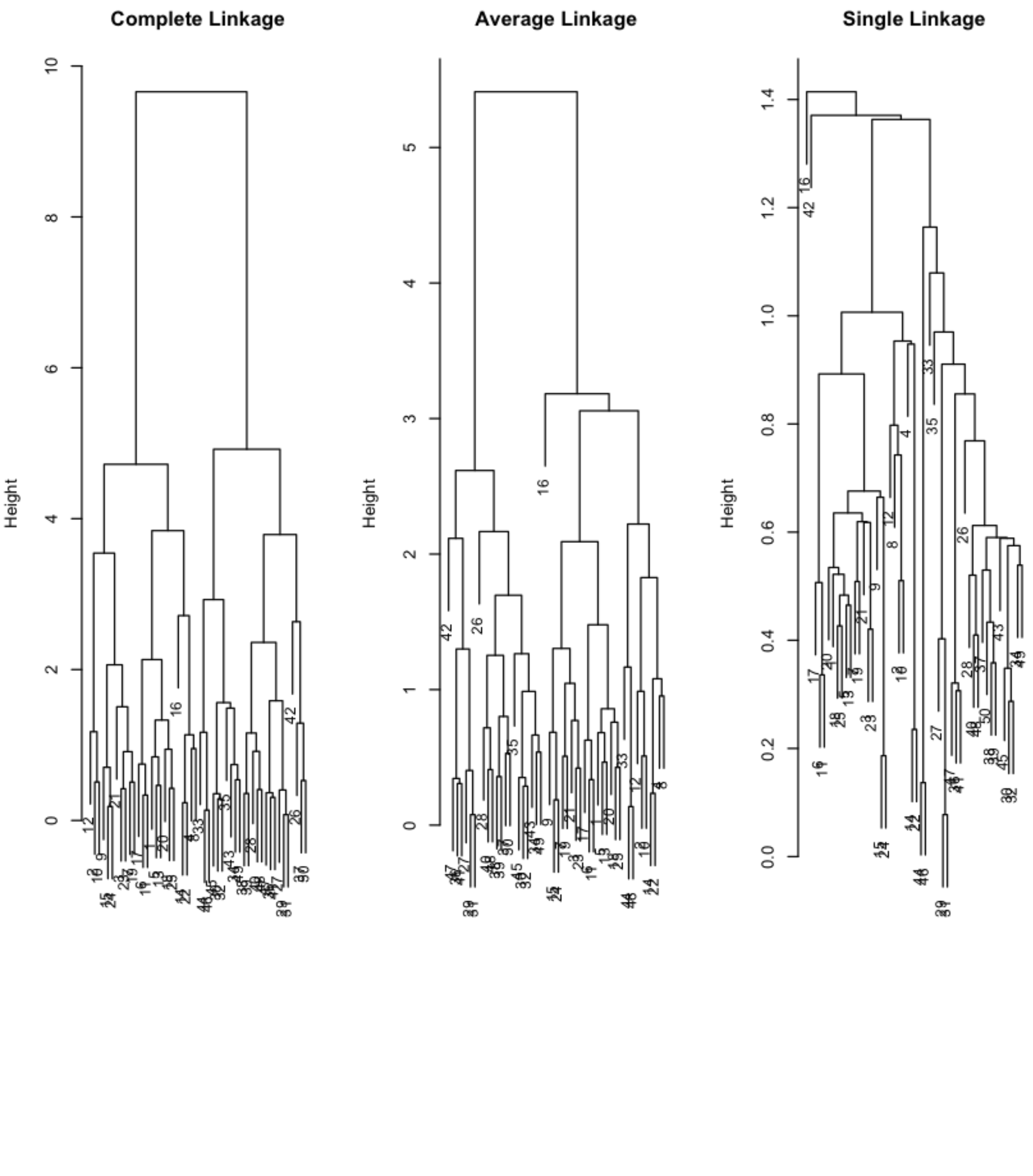
```
hc.complete=hclust(dist(x), method="complete")
```

In [37]:

```
hc.average=hclust(dist(x), method="average")
hc.single=hclust(dist(x), method="single")
```

In [38]:

```
par(mfrow=c(1,3))
plot(hc.complete,main="Complete Linkage", xlab="", sub="",
cex =.9)

plot(hc.average , main="Average Linkage", xlab="", sub="",
cex =.9)

plot(hc.single , main="Single Linkage", xlab="", sub="",
cex =.9)
```



In [39]:

```
cutree(hc.complete, 2)
```

```
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2
2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
```

In [40]:

```
cutree(hc.average, 2)
```

```
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2
2  2  2  2  2  1  2  2  2  2  2  2  2  2  2  2  1  2  1  2  2  2  2
```

In [41]:

```
cutree(hc.single, 2)
```

```
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

In [42]:

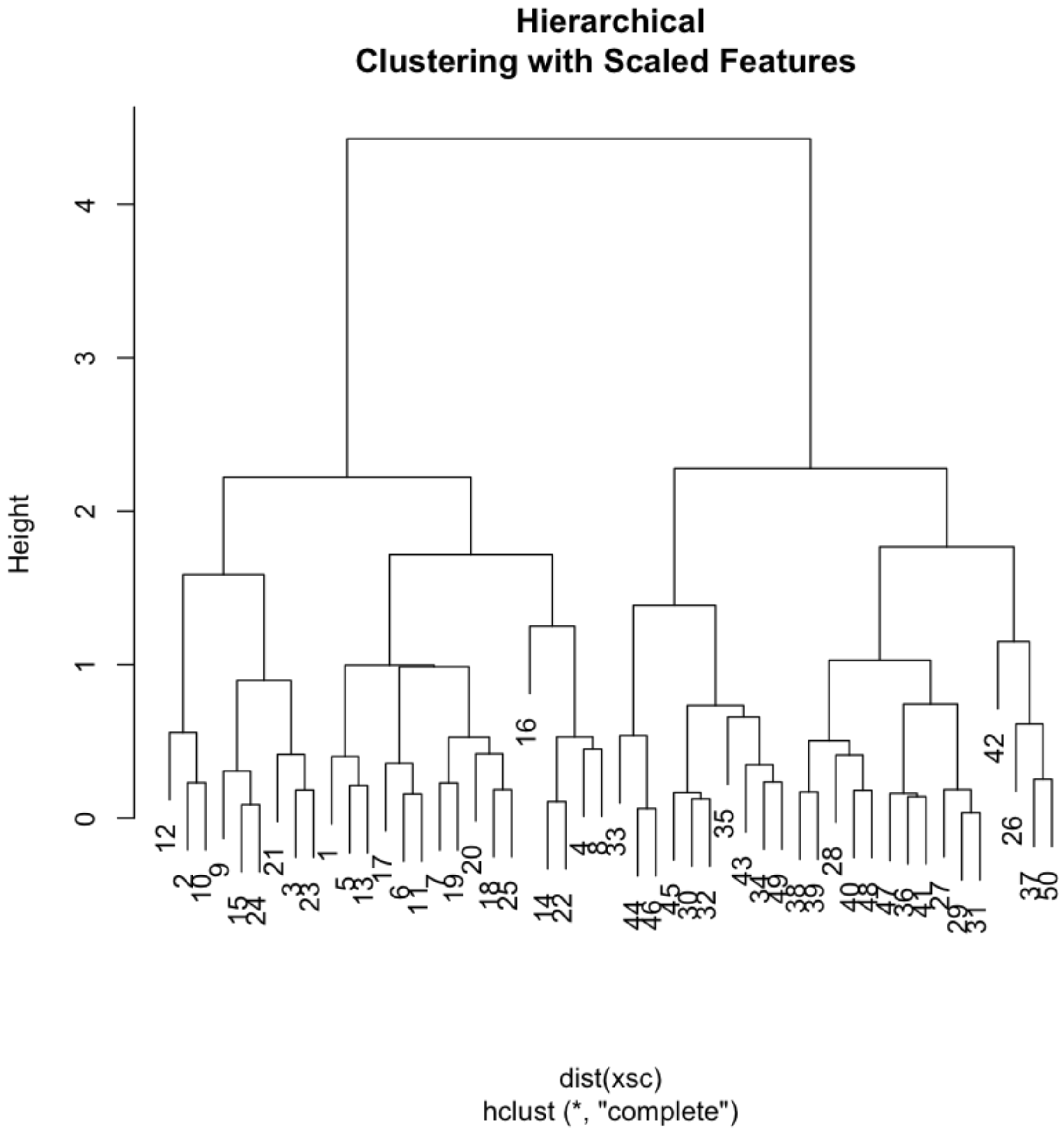
```
cutree(hc.single,4)
```

```
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  3  3
3  3  3  3  3  3  3  3  3  3  3  3  3  3  4  3  3  3  3  3  3  3  3
```

In [43]:

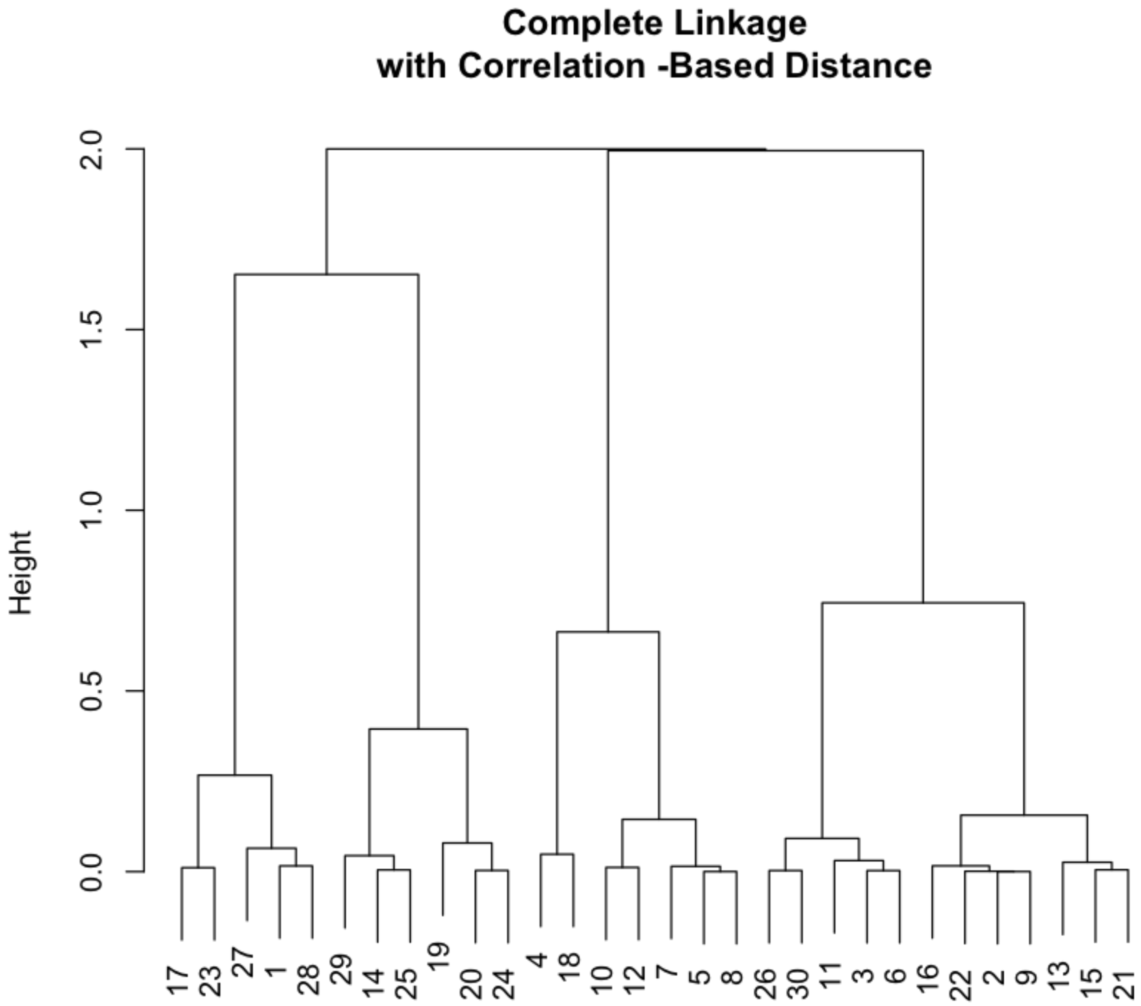
```
xsc=scale(x)
```

In [44]:
```
plot(hclust(dist(xsc), method="complete"), main="Hierarchical
Clustering with Scaled Features ")
```



In [45]:

```
x=matrix(rnorm(30*3), ncol=3)
dd=as.dist(1-cor(t(x)))
plot(hclust(dd, method="complete"), main="Complete Linkage
with Correlation -Based Distance", xlab="", sub="")
```



#Lab 3: NCI60 Data Example

In [46]:

```
library(ISLR)
```

In [47]:

```
nci.labs=NCI60$labs
nci.data=NCI60$data
```

In [48]:

```
dim(nci.data)
```

```
64  6830
```

In [49]:

```
nci.labs[1:4]
```

```
'CNS'  'CNS'  'CNS'  'RENAL'
```

In [50]:

```
table(nci.labs)
```

```
nci.labs
     BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEM
IA
          7           5           7           1           1          
6
MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTA
TE
          1           1           8           9           6          
2
      RENAL     UNKNOWN 
          9           1 
```

In [51]:

```
nci.labs
```

```
'CNS'  'CNS'  'CNS'  'RENAL'  'BREAST'  'CNS'  'CNS'  'BREAST'  'NSCLC'
'NSCLC'  'RENAL'  'RENAL'  'RENAL'  'RENAL'  'RENAL'  'RENAL'  'RENAL'
'BREAST'  'NSCLC'  'RENAL'  'UNKNOWN'  'OVARIAN'  'MELANOMA'  'PROSTATE'
'OVARIAN'  'OVARIAN'  'OVARIAN'  'OVARIAN'  'OVARIAN'  'PROSTATE'  'NSCLC'
'NSCLC'  'NSCLC'  'LEUKEMIA'  'K562B-repro'  'K562A-repro'  'LEUKEMIA'
'LEUKEMIA'  'LEUKEMIA'  'LEUKEMIA'  'LEUKEMIA'  'COLON'  'COLON'  'COLON'
'COLON'  'COLON'  'COLON'  'COLON'  'MCF7A-repro'  'BREAST'  'MCF7D-repro'
'BREAST'  'NSCLC'  'NSCLC'  'NSCLC'  'MELANOMA'  'BREAST'  'BREAST'
'MELANOMA'  'MELANOMA'  'MELANOMA'  'MELANOMA'  'MELANOMA'
'MELANOMA'
```

In [52]:

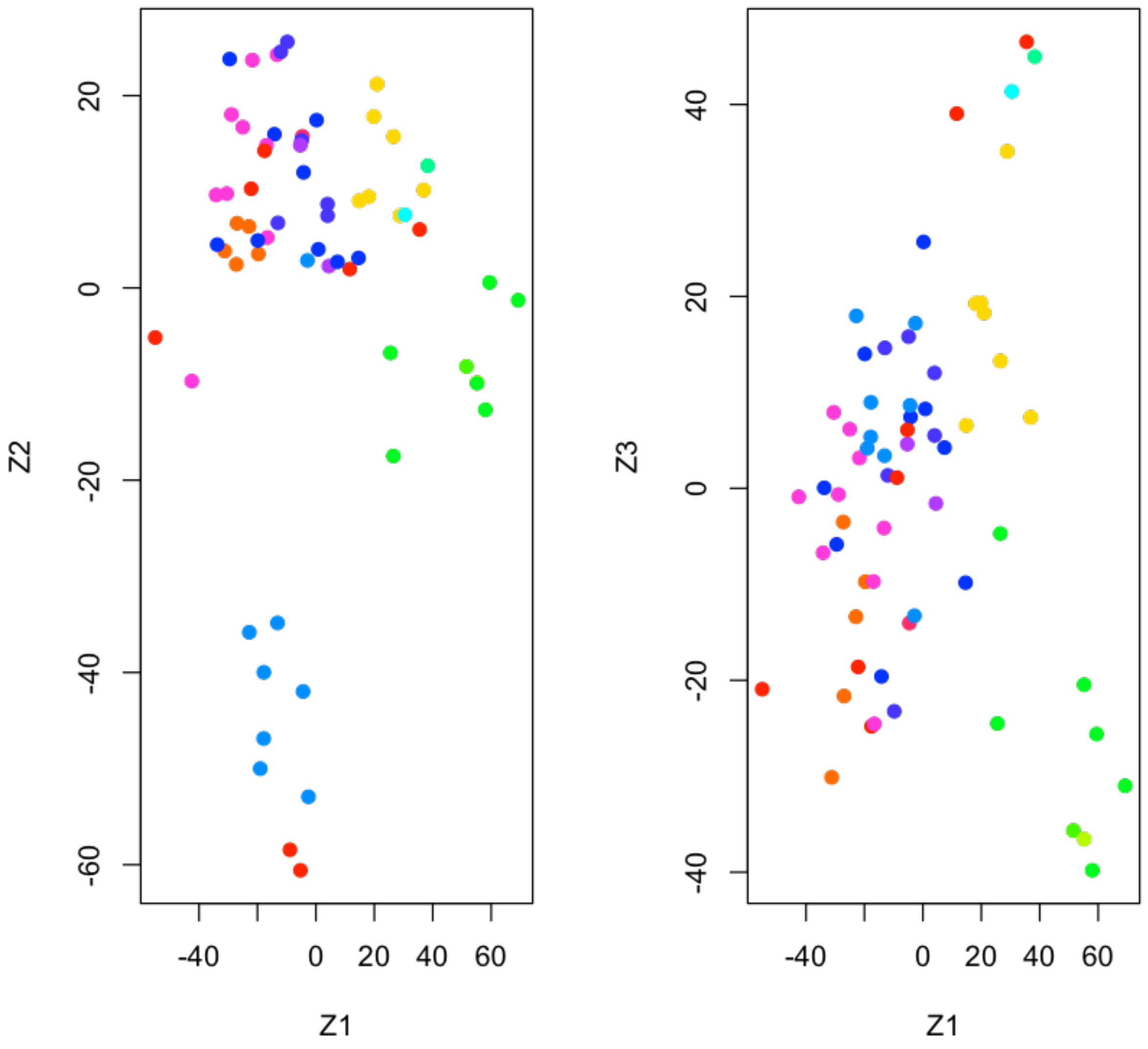
```
#PCA on NC160 data
pr.out=prcomp(nci.data, scale=TRUE)
```

In [53]:

```
Cols=function(vec){
cols=rainbow(length(unique(vec)))
return(cols[as.numeric(as.factor(vec))])
}
```

In [54]:

```
par(mfrow=c(1,2))
plot(pr.out$x[,1:2], col=Cols(nci.labs), pch=19,
xlab="Z1",ylab="Z2")

plot(pr.out$x[,c(1,3)], col=Cols(nci.labs), pch=19,
xlab="Z1",ylab="Z3")
```



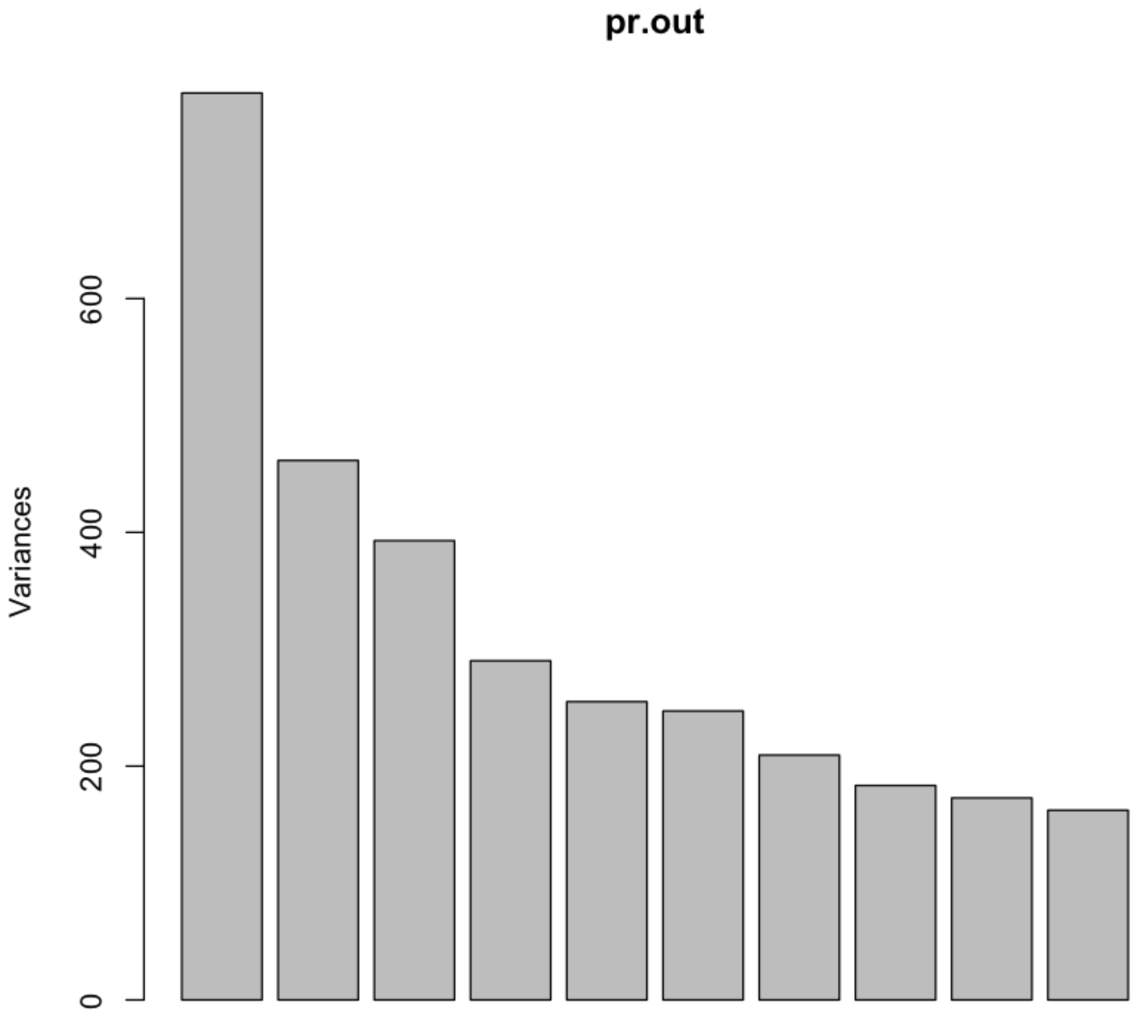
In [55]:

```
summary(pr.out)
```

```
Importance of components:
                           PC1      PC2      PC3      PC4      PC5
PC6
Standard deviation     27.8535 21.48136 19.82046 17.03256 15.97181 15
.72108
Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735  0
Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735  0
.03619
Cumulative Proportion   0.1136  0.18115  0.23867  0.28115  0.31850  0
.35468
                            PC7      PC8      PC9     PC10     PC11
PC12
Standard deviation     14.47145 13.54427 13.14400 12.73860 12.68672 1
2.15769
Proportion of Variance  0.03066  0.02686  0.02529  0.02376  0.02357
0.02164
Cumulative Proportion   0.38534  0.41220  0.43750  0.46126  0.48482
0.50646
                           PC13     PC14     PC15     PC16     PC17
PC18
Standard deviation     11.83019 11.62554 11.43779 11.00051 10.65666 1
0.48880
Proportion of Variance  0.02049  0.01979  0.01915  0.01772  0.01663
0.01611
Cumulative Proportion   0.52695  0.54674  0.56590  0.58361  0.60024
0.61635
                           PC19    PC20     PC21    PC22    PC23    P
C24
Standard deviation     10.43518 10.3219 10.14608 10.0544 9.90265 9.64
766
Proportion of Variance  0.01594  0.0156  0.01507  0.0148 0.01436 0.01
363
Cumulative Proportion   0.63229  0.6479  0.66296  0.6778 0.69212 0.70
575
                          PC25    PC26    PC27   PC28    PC29    PC30
PC31
Standard deviation     9.50764 9.33253 9.27320 9.0900 8.98117 8.75003
8.59962
Proportion of Variance 0.01324 0.01275 0.01259 0.0121 0.01181 0.01121
0.01083
Cumulative Proportion  0.71899 0.73174 0.74433 0.7564 0.76824 0.77945
0.79027
                          PC32    PC33    PC34    PC35    PC36    PC3
7    PC38
Standard deviation     8.44738 8.37305 8.21579 8.15731 7.97465 7.9044
6 7.82127
Proportion of Variance 0.01045 0.01026 0.00988 0.00974 0.00931 0.0091
5 0.00896
Cumulative Proportion  0.80072 0.81099 0.82087 0.83061 0.83992 0.8490
7 0.85803
                          PC39    PC40    PC41   PC42    PC43   PC44
PC45
Standard deviation     7.72156 7.58603 7.45619 7.3444 7.10449 7.0131
6.95839
Proportion of Variance 0.00873 0.00843 0.00814 0.0079 0.00739 0.0072
0.00709
Cumulative Proportion  0.86676 0.87519 0.88333 0.8912 0.89861 0.9058
Cumulative Proportion  0.86676 0.87518 0.88332 0.8912 0.89861 0.9058
0.91290
                         PC46    PC47    PC48    PC49    PC50    PC51
PC52
Standard deviation     6.8663 6.80744 6.64763 6.61607 6.40793 6.21984
6.20326
Proportion of Variance 0.0069 0.00678 0.00647 0.00641 0.00601 0.00566
0.00563
Cumulative Proportion  0.9198 0.92659 0.93306 0.93947 0.94548 0.95114
0.95678
                          PC53    PC54    PC55    PC56    PC57   PC58
PC59
Standard deviation     6.06706 5.91805 5.91233 5.73539 5.47261 5.2921
5.02117
Proportion of Variance 0.00539 0.00513 0.00512 0.00482 0.00438 0.0041
0.00369
Cumulative Proportion  0.96216 0.96729 0.97241 0.97723 0.98161 0.9857
0.98940
                          PC60    PC61    PC62    PC63      PC64
Standard deviation     4.68398 4.17567 4.08212 4.04124 1.237e-14
Proportion of Variance 0.00321 0.00255 0.00244 0.00239 0.000e+00
Cumulative Proportion  0.99262 0.99517 0.99761 1.00000 1.000e+00
```

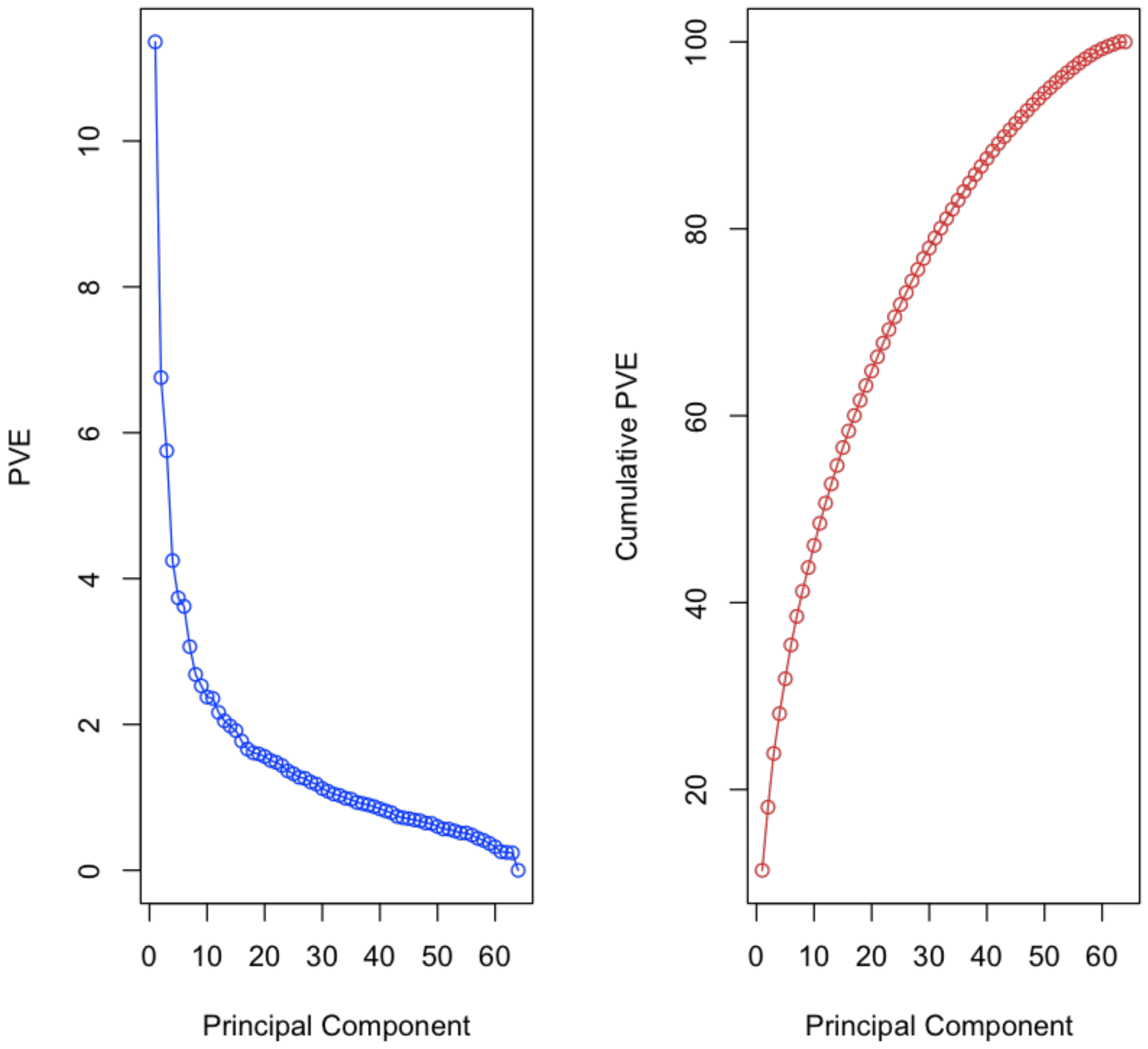
In [56]:

```
plot(pr.out)
```



In [57]:

```
pve=100*pr.out$sdev^2/sum(pr.out$sdev^2)
par(mfrow=c(1,2))
plot(pve, type="o", ylab="PVE", xlab="Principal Component",
col =" blue ")
plot(cumsum(pve), type="o", ylab="Cumulative PVE", xlab="
Principal Component ", col =" brown3 ")
```

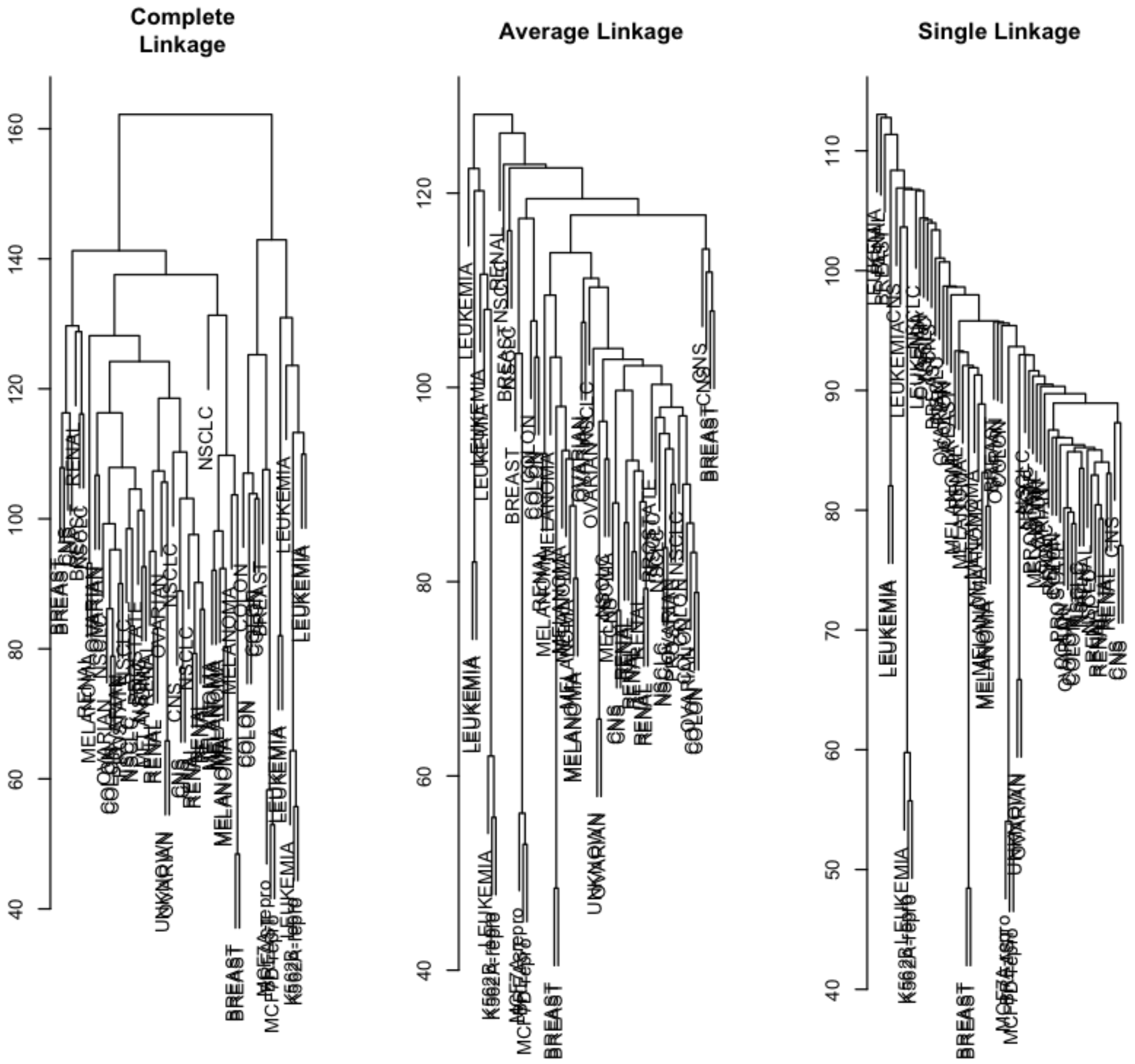


In [58]:

```
#Clustering the Observations of the NCI60 Data
sd.data=scale(nci.data)
```

In [59]:

```
par(mfrow=c(1,3))
data.dist=dist(sd.data)
plot(hclust(data.dist), labels=nci.labs, main="Complete
Linkage", xlab="", sub="",ylab="")
plot(hclust(data.dist, method="average"), labels=nci.labs,
main="Average Linkage", xlab="", sub="",ylab="")
plot(hclust(data.dist, method="single"), labels=nci.labs,
main="Single Linkage", xlab="", sub="",ylab="")
```



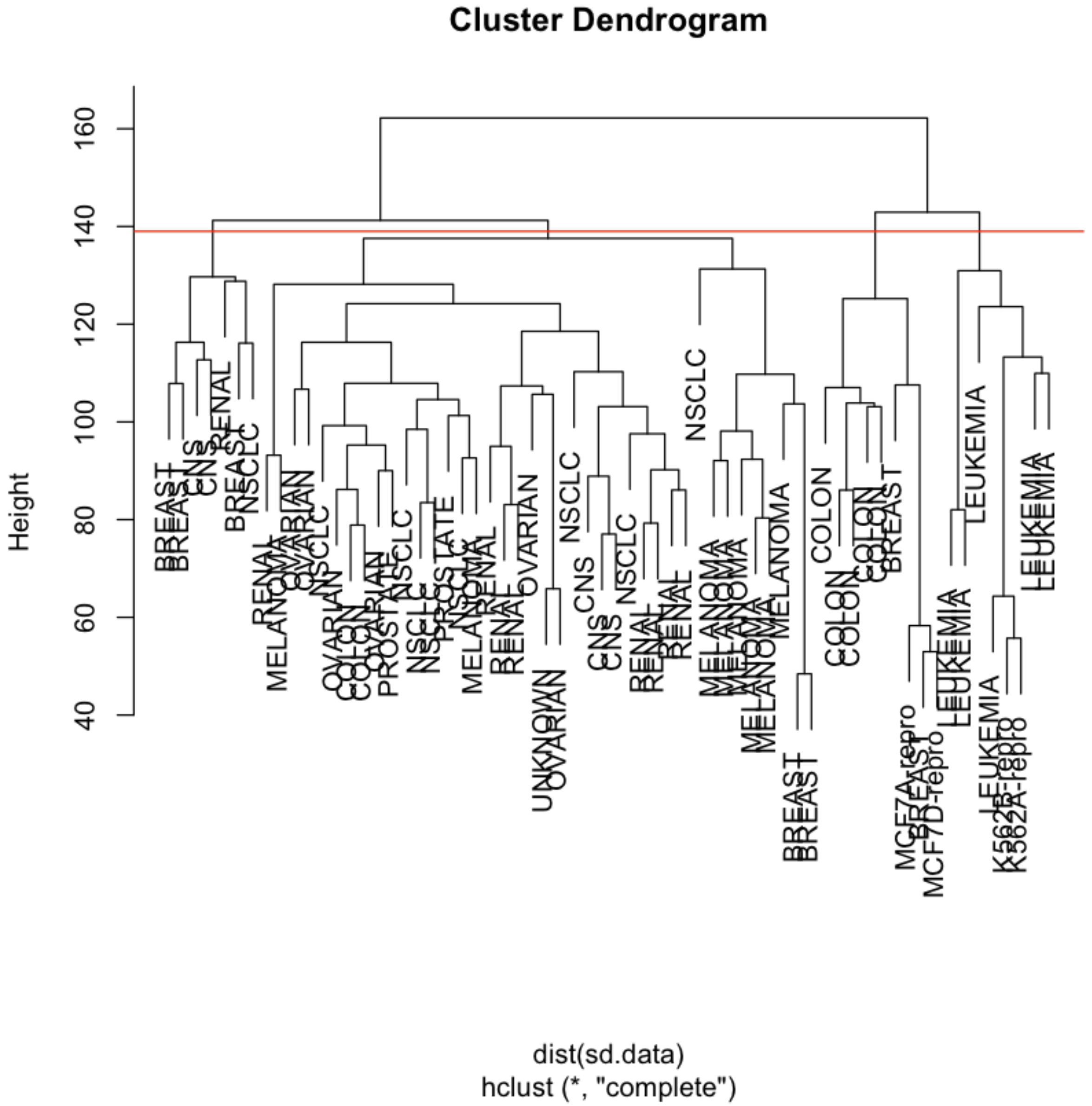
In [60]:

```
hc.out=hclust(dist(sd.data))
hc.clusters=cutree(hc.out,4)
table(hc.clusters,nci.labs)
```

```
           nci.labs
hc.clusters BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-r
epro
          1      2   3     2           0           0        0
0
          2      3   2     0           0           0        0
0
          3      0   0     0           1           1        6
0
          4      2   0     5           0           0        0
1
           nci.labs
hc.clusters MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
          1           0        8     8       6        2     8       1
          2           0        0     1       0        0     1       0
          3           0        0     0       0        0     0       0
          4           1        0     0       0        0     0       0
```

In [61]:

```
par(mfrow=c(1,1))
plot(hc.out, labels=nci.labs)
abline(h=139, col="red")
```



In [62]:

```
hc.out
```

```
Call:
hclust(d = dist(sd.data))

Cluster method   : complete 
Distance         : euclidean 
Number of objects: 64 
```

In [65]:

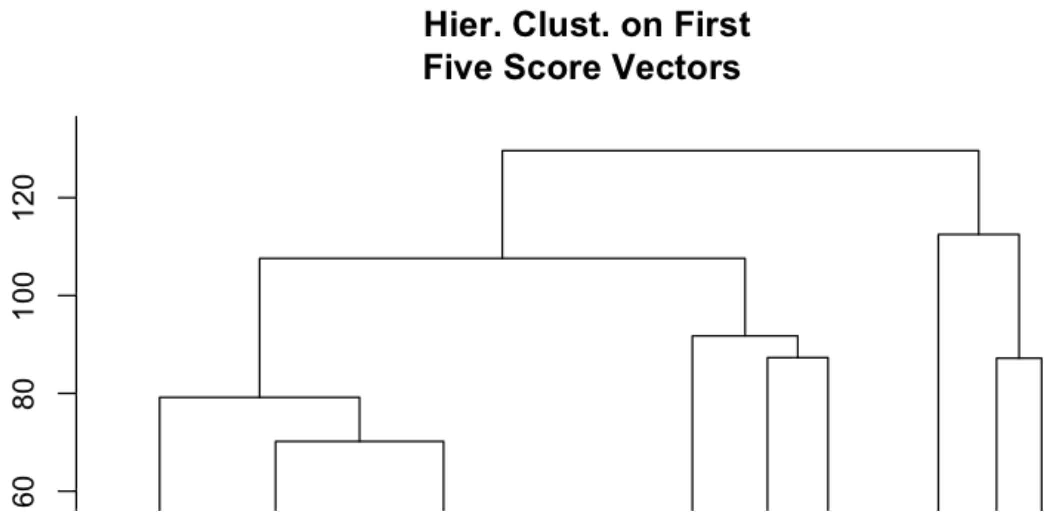
```
set.seed(4)
km.out=kmeans(sd.data, 4, nstart=20)
km.clusters=km.out$cluster
table(km.clusters ,hc.clusters)
```

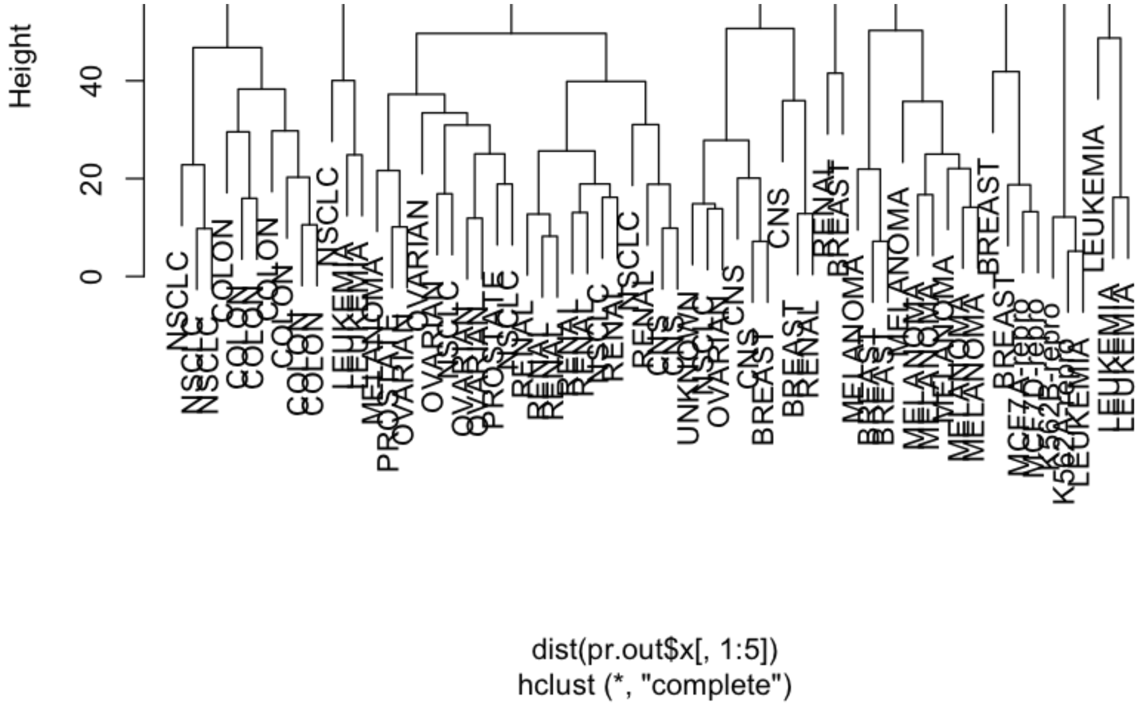
```
           hc.clusters
km.clusters  1  2  3  4
          1 11  0  0  9
          2  9  0  0  0
          3  0  0  8  0
          4 20  7  0  0
```

In [66]:

```
hc.out=hclust(dist(pr.out$x[,1:5]))
plot(hc.out, labels=nci.labs, main="Hier. Clust. on First
Five Score Vectors ")
table(cutree(hc.out,4), nci.labs)
```

```
   nci.labs
    BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro MCF
7D-repro
  1      0   2     7           0           0        2           0
0
  2      5   3     0           0           0        0           0
0
  3      0   0     0           1           1        4           0
0
  4      2   0     0           0           0        0           1
1
   nci.labs
    MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
  1        1     8       5        2     7       0
  2        7     1       1        0     2       1
  3        0     0       0        0     0       0
  4        0     0       0        0     0       0
```




Height
0
20
40
NSCLC
NSCLC
NSCLC
COLON
COLON
COLON
COLON
COLON
COLON
COLON
NSCLC
LEUKEMIA
LEUKEMIA
MELANOMA
PROSTATE
OVARIAN
OVARIAN
OVARIAN
NSCLC
OVARIAN
OVARIAN
PROSTATE
NSCLC
RENAL
RENAL
RENAL
RENAL
RENAL
NSCLC
RENAL
RENAL
NSCLC
RENAL
CNS
CNS
UNKNOWN
NSCLC
OVARIAN
CNS
CNS
BREAST
CNS
BREAST
RENAL
RENAL
BREAST
MELANOMA
BREAST
BREAST
MELANOMA
MELANOMA
MELANOMA
MELANOMA
MELANOMA
MELANOMA
BREAST
BREAST
MCF7A-repro
MCF7D-repro
K562B-repro
K562A-repro
LEUKEMIA
LEUKEMIA
LEUKEMIA
LEUKEMIA
dist(pr.out$x[, 1:5])
hclust (*, "complete")

In [ ]: