R Minicourse Workshop, Part 4

Presented to the
Washington State Deptment of Ecology
September 2–3, 2014

Dr. Robin Matthews, Institute for Watershed Studies
Dr. Geoffrey Matthews, Computer Science Department
Western Washington University

Initial Examination of Multivariate Data

- Begin by plotting and using simple exploratory tools like analysis to look for patterns
  - Don't use complicated multivariate tests to describe simple univariate or bivariate patterns
- Check for normality and homoscedasticity . . . nearly all multivariate methods are sensitive to heteroscedastic variances
- Identify redundant, nonlinear, and random variables
  - Including redundant, nonlinear, and random variables in multivariate analysis can obscure patterns in the remaining variables
- Identify variables with *zero* variance (all samples have same value)
  - This type of response isn't useful in multivariate analysis

Introduction to Multivariate Analysis, Continued

- Most multivariate methods involve reorganizing the data matrix to find linear or monotonic patterns, or simplifying a complex data sets to identify a subset of variables that best describe the patterns in the data
  - Not all multivariate patterns will be linear or monotonic!
  - Multivariate patterns can be significant even if the individual univariate patterns are not significant
- Two common multivariate patterns include similarity among groups of samples (clustering) and increasing dissimilarity along a gradient (ordination)

Introduction to Multivariate Analysis

Clustering vs. Ordination

Clustering involves finding similarity among groups of samples:

A	B	C	D	E	F	E	C	A	D	F	B
2	3	1	2	1	3	1	1	2	2	3	3
2	3	1	2	1	3	1	1	2	2	3	3
3	1	2	3	2	1	2	2	3	3	1	1
3	1	2	3	2	1	2	2	3	3	1	1
1	2	3	1	3	2	3	3	1	1	2	2
1	2	3	1	3	2	3	3	1	1	2	2

Ordination looks for increasing dissimilarity along a gradient:

A	B	C	D	E	F	E	C	A	D	F	B
3	6	2	4	1	5	1	2	3	4	5	6
2	5	1	3	6	4	6	1	2	3	4	5
1	4	6	2	5	3	5	6	1	2	3	4
6	3	5	1	4	2	4	5	6	1	2	3
5	2	4	6	3	1	3	4	5	6	1	2
4	1	3	5	2	6	2	3	4	5	6	1

Multivariate Ordination

Principal Components Analysis

- PCA is a linear model that searches for combinations of variables that explain the most variance in the data
- Because PCA is a linear model, it is influenced by all problems affecting regression/correlation
- PCA uses *all variables*, so random variables can be a problem
- PCA uses combinations of variables, so multivariate homoscedasticity is important
    - Most PCA applications are *row centered* and *standardized*, which converts from a co-variance PCA to a correlation PCA

Principal Components Analysis

R Syntax Using prcomp and princomp

- There are two basic PCA methods: princomp and prcomp
  - princomp ordinated using an eigenvalue matrix
  - prcomp is based on a singular value decomposition of the data matrix, which is generally preferred over princomp
  - princomp and prcomp will often produce identical results (number of principal components = number of variables)
  - But if there are a large number of variables, prcomp truncates after "almost all" of the variance is contained in the ordination (number of principal components < number of variables)
- Both default to a covariance matrix (matches S-Plus), but the best option is a scaled, centered correlation matrix[1]
- In both methods, omit variables that are constant (e.g., all zeros)

[1] Similar to standard normal distribution with $\sigma = 1$ and $\overline{x} = 0$

Principal Components Analysis - Iris Data

Comparison of princomp and prcomp

```
##### PRINCOMP VERSION WITH SCALED/CENTERED CORRELATION MATRIX
data(iris); attach(iris)
iris.princomp <- princomp(iris[, c(1:4)], cor=T) #Basic PCA command
summary(iris.princomp)
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4
Standard deviation     1.7083611 0.9560494 0.38308860 0.143926497
Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000
```

```
##### PRCOMP VERSION WITH SCALED/CENTERED CORRELATION MATRIX
iris.prcomp <- prcomp(iris[, c(1:4)], scale=T, center=T)
```

```
summary(iris.prcomp)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Principal Components Analysis - Iris Data

Examining Variable and Sample Ordinations in prcomp

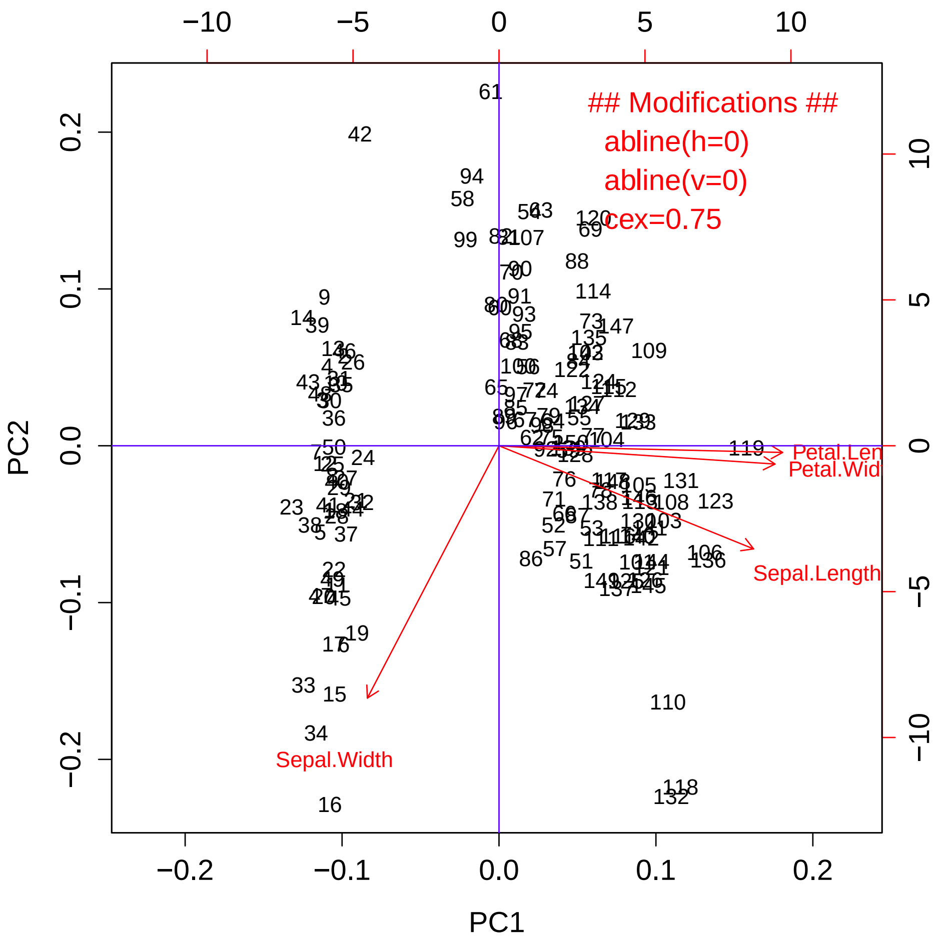
PCA produces sample ordinations (n=150) that show the location of each sample on PC1-PC4 and variable ordinations (n=4) that show the relationship (correlation) for each variable on PC1-PC4

```
 iris.prcomp$rotation  ### for princomp: iris.princomp$loading

                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

```
iris.prcomp$x          ### for princomp: iris.princomp$scores

               PC1          PC2          PC3          PC4
  [1,] -2.25714118 -0.478423832  0.127279624  0.024087508
  [2,] -2.07401302  0.671882687  0.233825517  0.102662845
  [3,] -2.35633511  0.340766425 -0.044053900  0.028282305
  .
  .
  .
[148,]  1.51609145 -0.268170747 -0.179576781  0.118773236
[149,]  1.36820418 -1.007877934 -0.930278721  0.026041407
[150,]  0.95744849  0.024250427 -0.526485033 -0.162533529
```

Principal Components Analysis - Iris Data

Using biplot to Show Sample and Variable Loading



```
## Modifications ##
abline(h=0)
abline(v=0)
cex=0.75
```

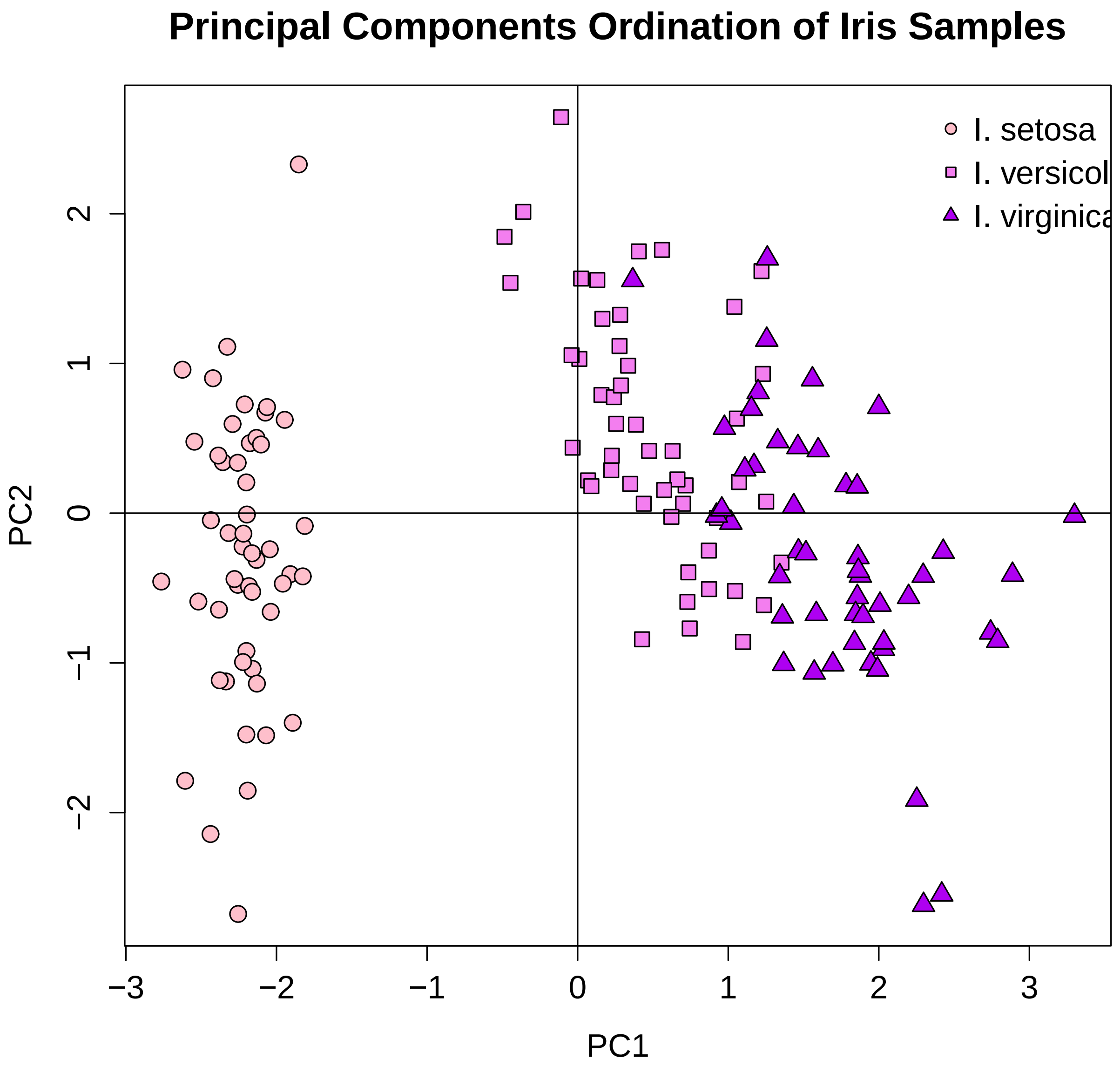
You can quickly examine sample and variable loading using the command `biplot(iris.prcomp)`. The numbers indicate rows (iris samples) and the arrows show the influence of each variable.

Plotting PCA Results on a Scatterplot

- Scatterplots are a common way to show sample or variable ordinations (see Part 2 for help with R plotting syntax)
- plot(iris.prcomp$x) plots the first two principal components and is equivalent to plot(iris.prcomp$x[,c(1:2)])
- The following example uses several advanced plotting features to create a scatterplot of the samples on PC1 and PC2

```
op = par(mfrow=c(1,1))
plot(iris.prcomp$x,
  main="Principal Components Ordination of Iris Samples",
  pch=c(21,22,24)[unclass(iris$Species)],
  bg=c("pink", "violet", "purple")[unclass(iris$Species)],
  cex=1.5)
abline(h=0); abline(v=0)
legend(x="topright", c("I. setosa", "I. versicolor", "I. virginica"),
       pch=c(21, 22, 24), pt.bg=c("pink", "violet", "purple"),
       bty="n", cex=1)

#### To plot PC2 and PC3:
plot(iris.prcomp$x[,c(2:3)])
```

Scatterplot Showing PCA Sample Ordination



Multivariate Analysis - Clustering

Hierarchical vs. Divisive Clustering

- Most commonly used technique is agglomerative, hierarchical clustering
  - Similarity (distance) is calculated between all samples
  - The two closest samples (most similar) are combined into a joined sample containing all data from the two original samples
  - The distances between all remaining samples and the joined sample are recalculated
  - The next two closest samples are joined, and so on
- Another common technique is divisive clustering, where initial groups are defined, then all clusters are iteratively regrouped until the with-in group distances are minimized

Multivariate Analysis - Clustering

Interpreting Clustering Output

- Clustering is primarily an exploratory data analysis tool
- Most clustering techniques are uninformed (you don't identify a grouping variable like site)
  - Divisive clustering requires that you specify the number of clusters to create, but the clusters are determined by similarity in the measured variables, not your definition of a group
  - This feature may identify groups you didn't expect or show you that your definition of a group is not correct
- You can cluster random data . . . there is no automatic significance test to prevent this from happening
  - You can test significance after clusters are identified

Hierarchical Clustering

Measuring Distance Between Samples

- The first step in hierarchical clustering is to calculate the distance between samples (`dist`)
- Some of the distance methods available in `R` include

Distance Method	`R` Syntax	Equation/Approach
Euclidean (Squared Euclidean)	`method="euclidean"`	$\sum (x_i - y_i)^2$
Maximum (Chebychev)	`method="maximum"`	$\max \lvert x_i - y_i \rvert$
Manhattan (City Block)	`method="manhattan"`	$\sum \lvert x_i - y_i \rvert$
Canberra	`method="canberra"`	$\sum \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Binary	`method="binary"`	count nonzero/zero

- The default is `method="euclidean"`

Hierarchical Clustering

Example of Squared Euclidean Distance Calculations

	Var. A	Var. B	Var. C	$X_i - Y_i$			
Site 1	20	10	17	Site 1 – Site 2:	5	10	17
Site 2	15	0	0	Site 1 – Site 3:	20	4	17
Site 3	0	6	0	Site 2 – Site 3:	15	–6	0

$$\text{Distance}_{\,1-2} = (20-15)^2 + (10-0)^2 + (17-0)^2 = (5^2 + 10^2 + 17^2) = 414$$

$$\text{Distance}_{\,1-3} = (20-0)^2 + (10-6)^2 + (17-0)^2 = (20^2 + 4^2 + 17^2) = 705$$

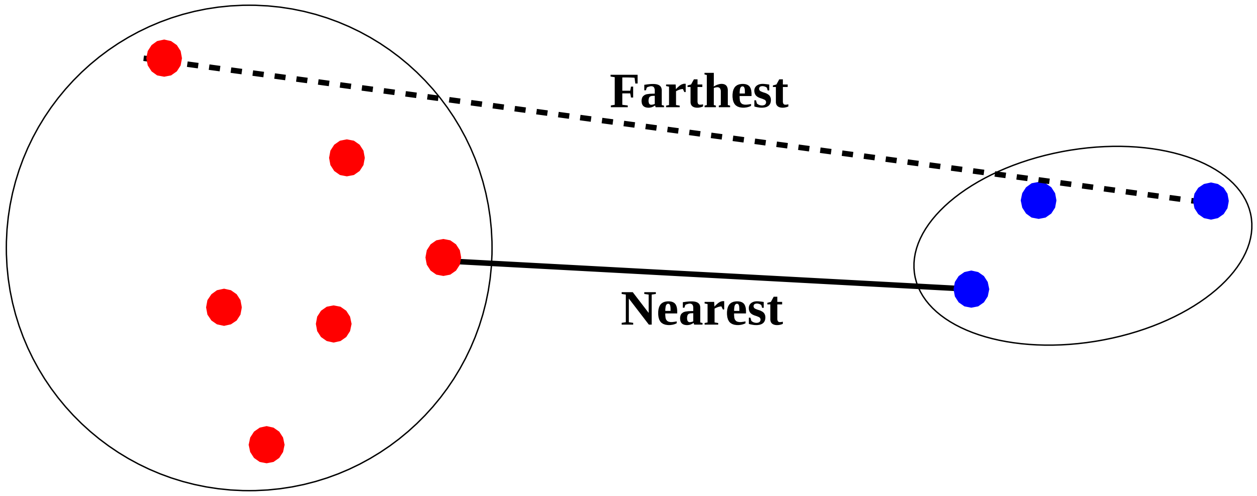
$$\text{Distance}_{\,2-3} = (15-0)^2 + (0-6)^2 + (0-0)^2 = (15^2 + -6^2 + 0^2) = 261$$

Sites 2 and 3 are the closest based on squared Euclidean distance

Hierarchical Clustering

Measuring Distances Between Joined Samples (Cluster Method)

- The results from `dist` are used with `hclust` to complete the iterative clustering process
- As with distance metrics, we can choose from a variety of clustering methods
- The default method is the farthest neighbor (complete linkage), which joins groups using the two most distant members of each cluster.



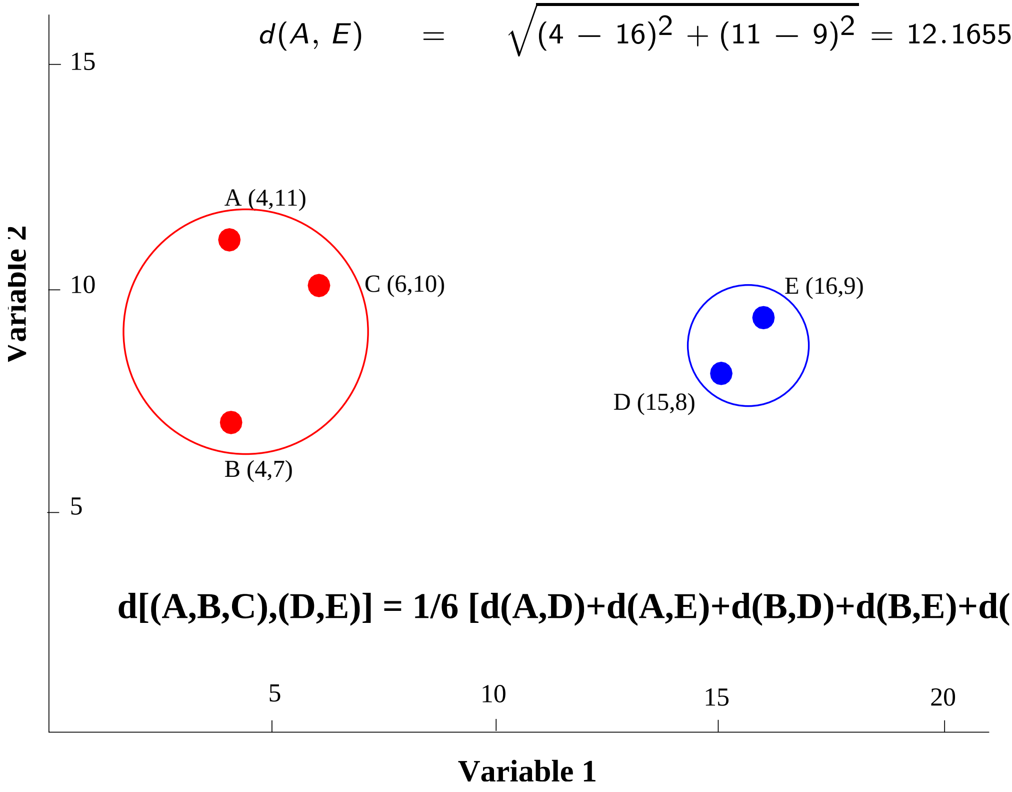
Hierarchical Clustering

Other Clustering Methods - Average Distance

Average distance (unweighted pair group method with arithmetic mean; UPGMA) gives equal weight to each sample in the cluster

$$d(A, D) = \sqrt{(4 - 15)^2 + (11 - 8)^2} = 11.4018$$

$$d(A, E) = \sqrt{(4 - 16)^2 + (11 - 9)^2} = 12.1655$$

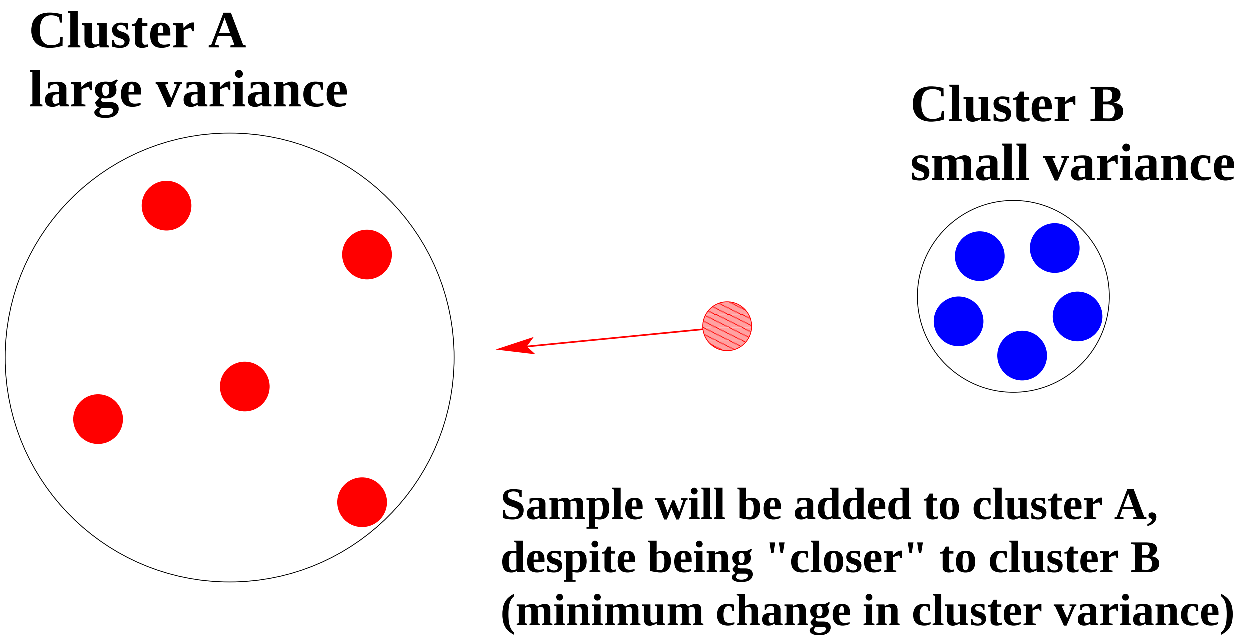


d[(A,B,C),(D,E)] = 1/6 [d(A,D)+d(A,E)+d(B,D)+d(B,E)+d(C,D)+d(C,E)]

Hierarchical Clustering

Other Clustering Methods - Wards Minimum Variance

- *Ward's minimum variance* often produces different results
- After each cluster cycle, the sample pairs with the lowest within-cluster sums of squares is joined next
- This approach preserves groups with small internal variance



Cluster A
large variance

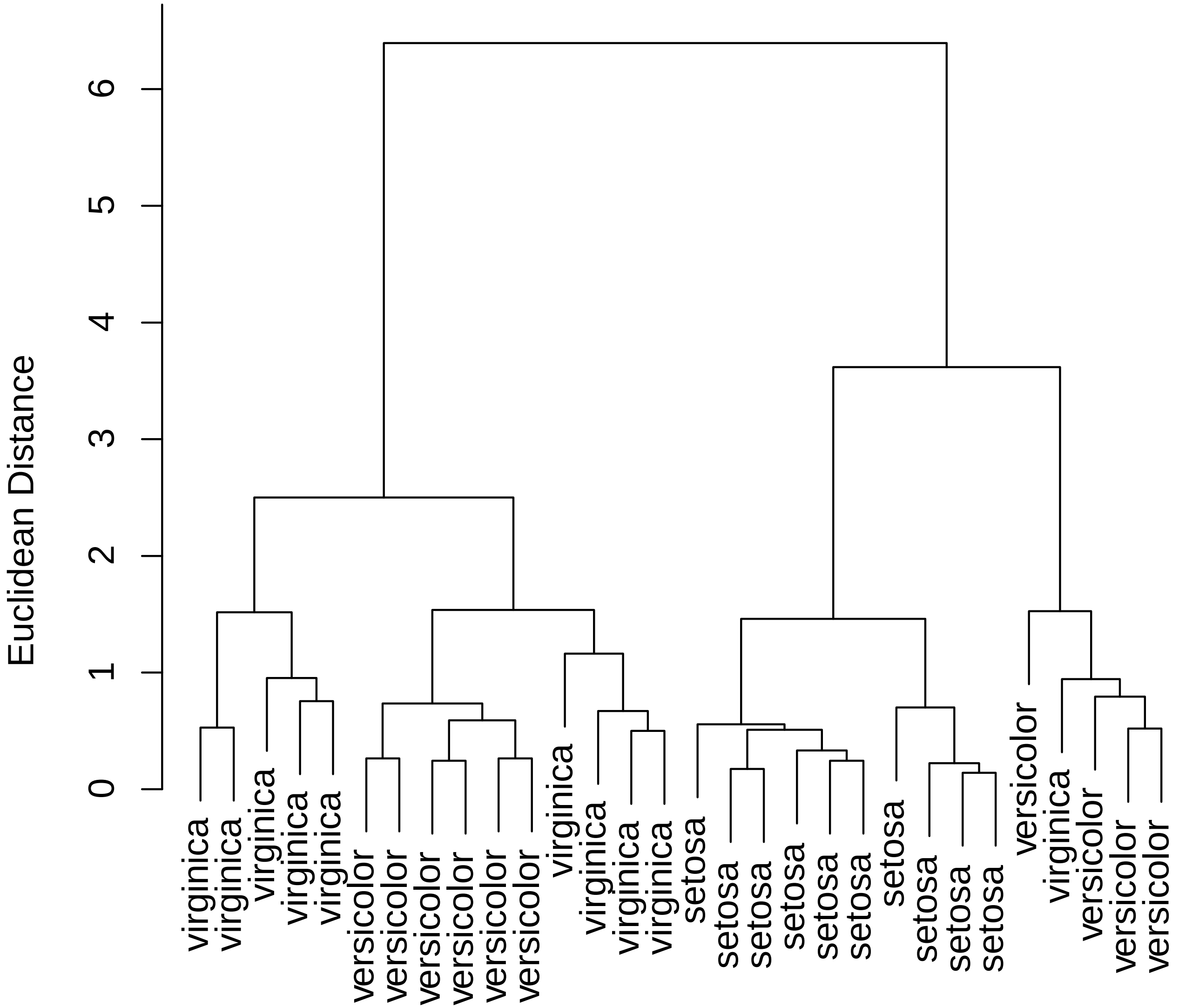
Cluster B
small variance

Sample will be added to cluster A,
despite being "closer" to cluster B
(minimum change in cluster variance)

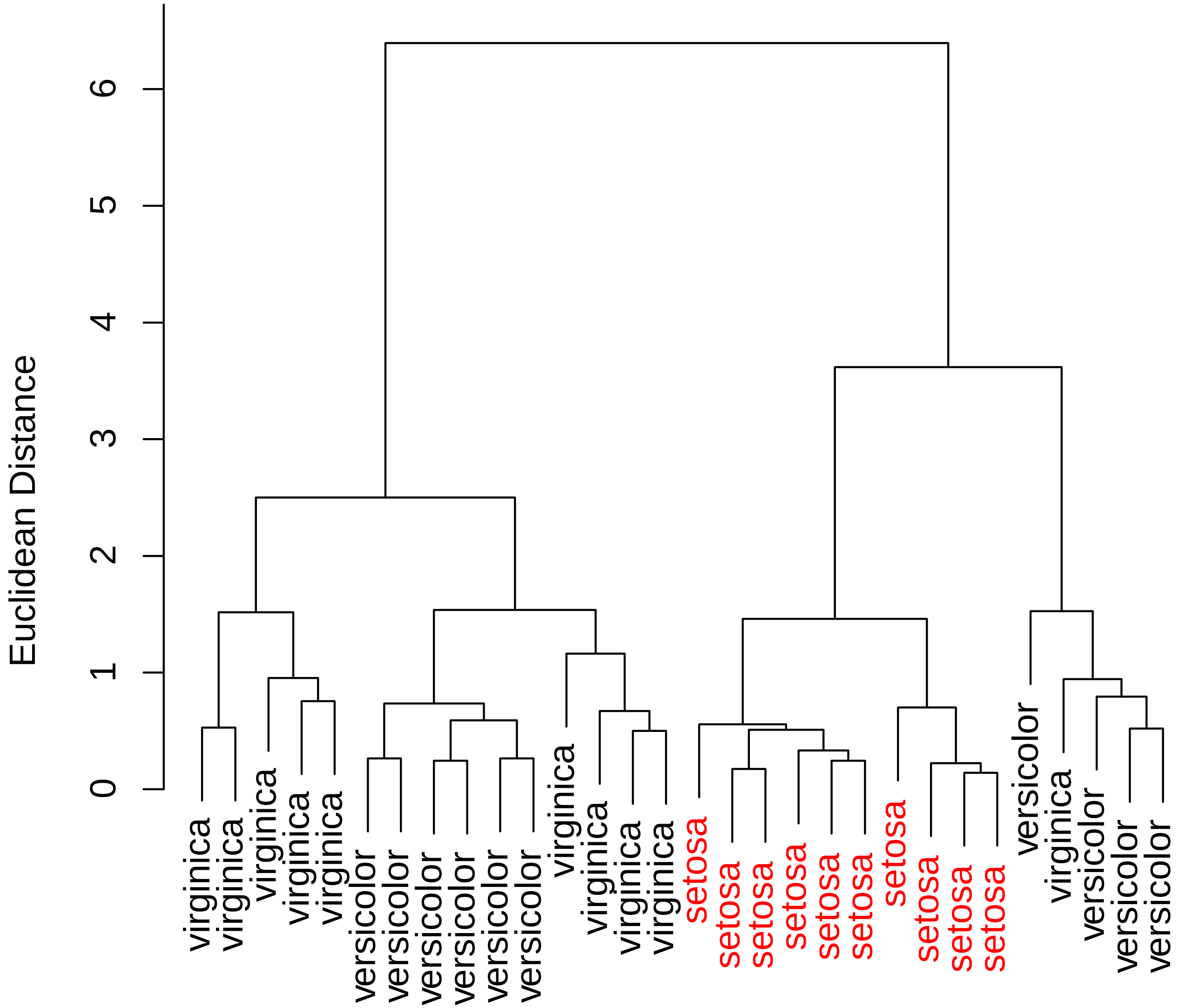
Clustering of First 10 Rows For Each Iris Species

```
### First example - Euclidean distance with farthest neighbor:
data(iris); attach(iris)

### Step 1:  select data subset and distance metric
edist <- dist(iris[c(1:10, 51:60, 101:110), c(1:4)],
        method="euclidean")

### Step 2: select clustering method
edist.complete <- hclust(edist, method="complete")

### Step 3: plot the results as an edited dendrogram
plot(edist.complete, labels=iris[c(1:10, 51:60, 101:110), 5],
     ylab="Euclidean Distance", xlab=" ", main=" ", sub=" ")


### Second example - Euclidean distance with Ward’s minimum variance

edist.ward <- hclust(edist, method="ward")

### Add hang=-1 to place samples on x-axis
plot(edist.ward, labels=iris[c(1:10, 51:60, 101:110), 5],
     ylab="Euclidean Distance", xlab=" ", main=" ", sub=" ", hang=-1)
```

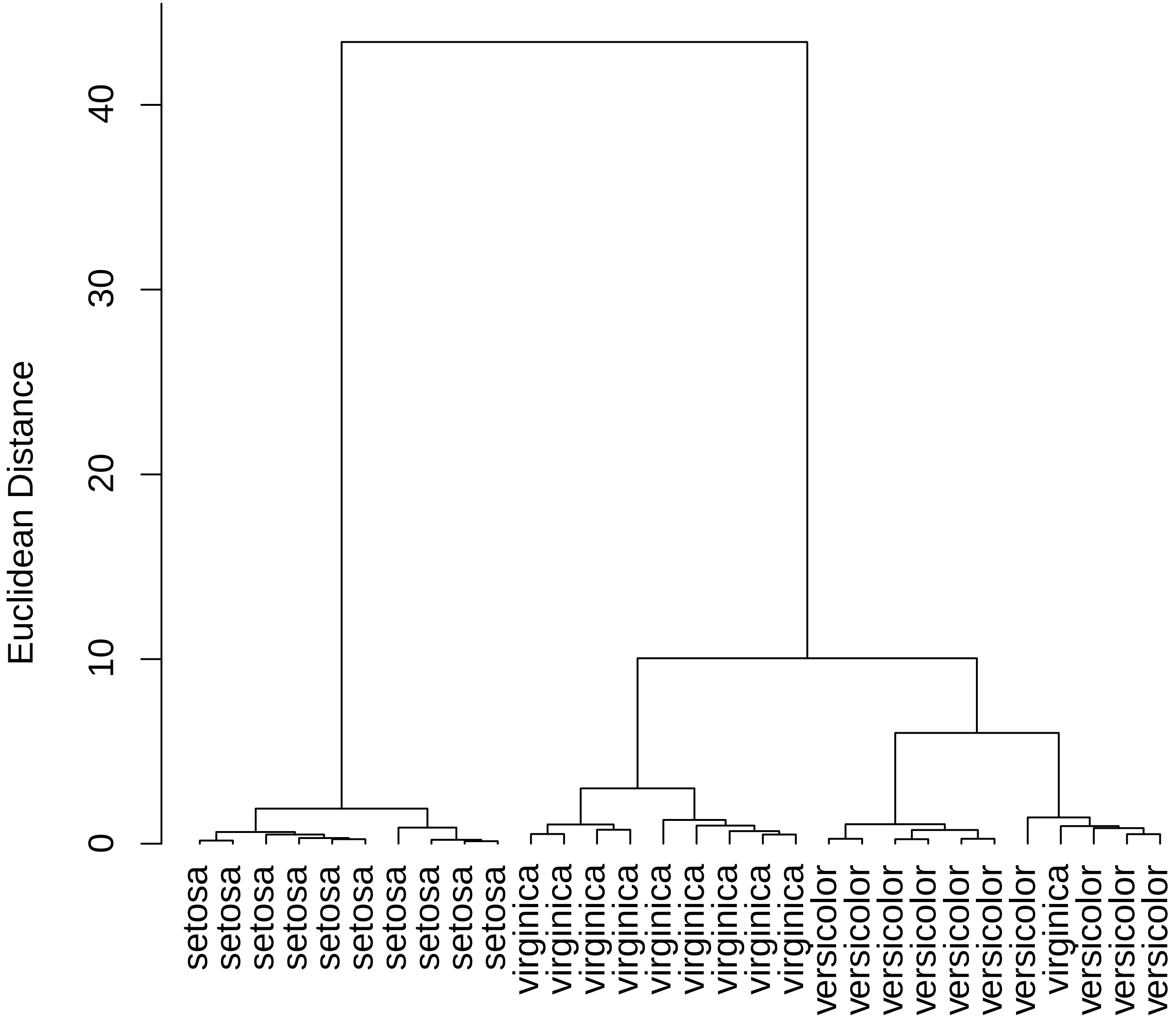
Euclidean Distance/Farthest Neighbor



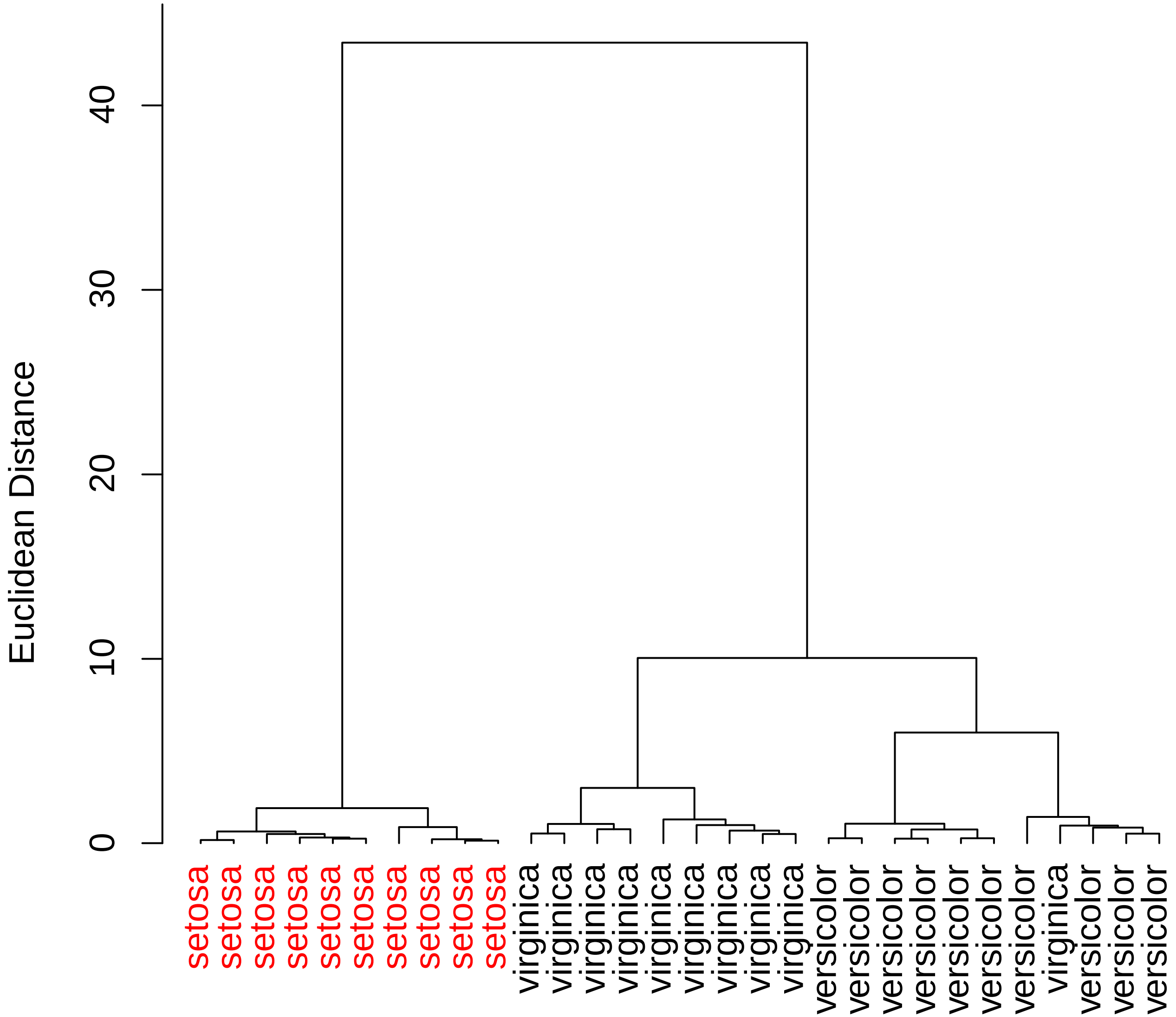
Euclidean Distance/Farthest Neighbor



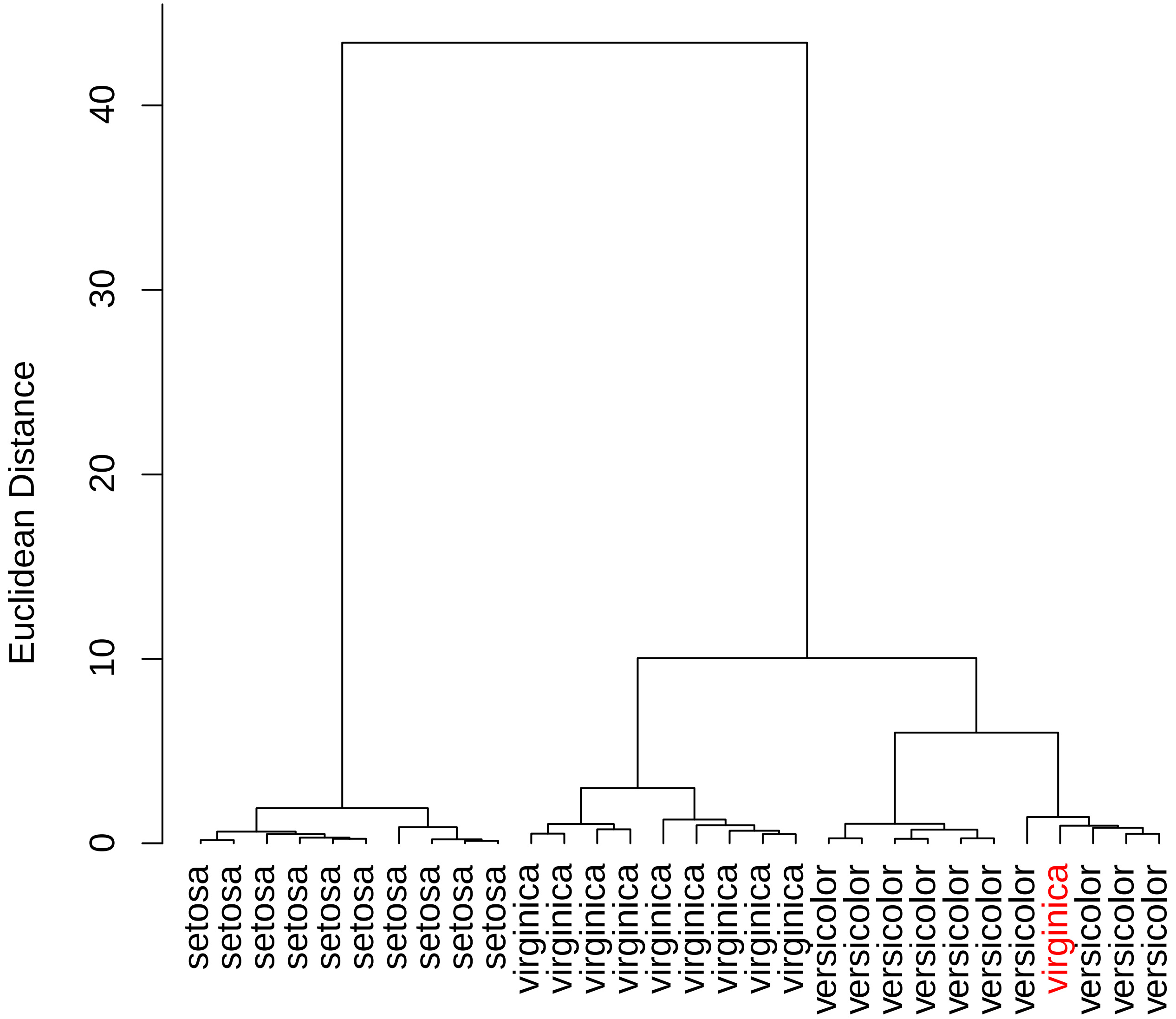
Euclidean Distance/Ward's Minimum Variance



Euclidean Distance/Ward's Minimum Variance



Euclidean Distance/Ward's Minimum Variance



Multivariate Analysis - Divisive Clustering

KMeans Clustering of Fisher's Iris Data (n=30)

- KMeans clustering starts with *n* centers for each variable
- Distances are computed to all points simultaneously
- The center is moved and distances recomputed, with the objective of minimizing distance (measured as within groups sums of squares)
- The R program lets you set the number of iterations or times the centers are moved. The default is 10 iterations
    - More iterations take time, but produce a more repeatable result
- Because the cluster centers are based on iterations, repeated k-means clustering can produce different results.

Divisive Clustering - Iris Data

R Syntax for KMeans Clustering into Three Groups

KMeans clustering produces the number of groups you request. In this example, we ask for 3 groups to match our assumption that each of the 3 species will cluster separately

```
irispart <- iris[c(1:10, 51:60, 101:110),] # R shortcut!
kcluster3 <- kmeans(irispart[ , c(1:4)], 3)
kcluster3  #This produces the cluster summary

### Edited output:
Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1        4.860       3.310     1.450000        0.22
2        6.875       3.025     6.012500        2.10
3        5.975       2.825     4.441667        1.45

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  51  52  53  54  55  56  57  58  59  60
  1   1   1   1   1   1   1   1   1   1   3   3   3   3   3   3   3   3   3   3
101 102 103 104 105 106 107 108 109 110
  2   3   2   2   2   2   3   2   2   2
```

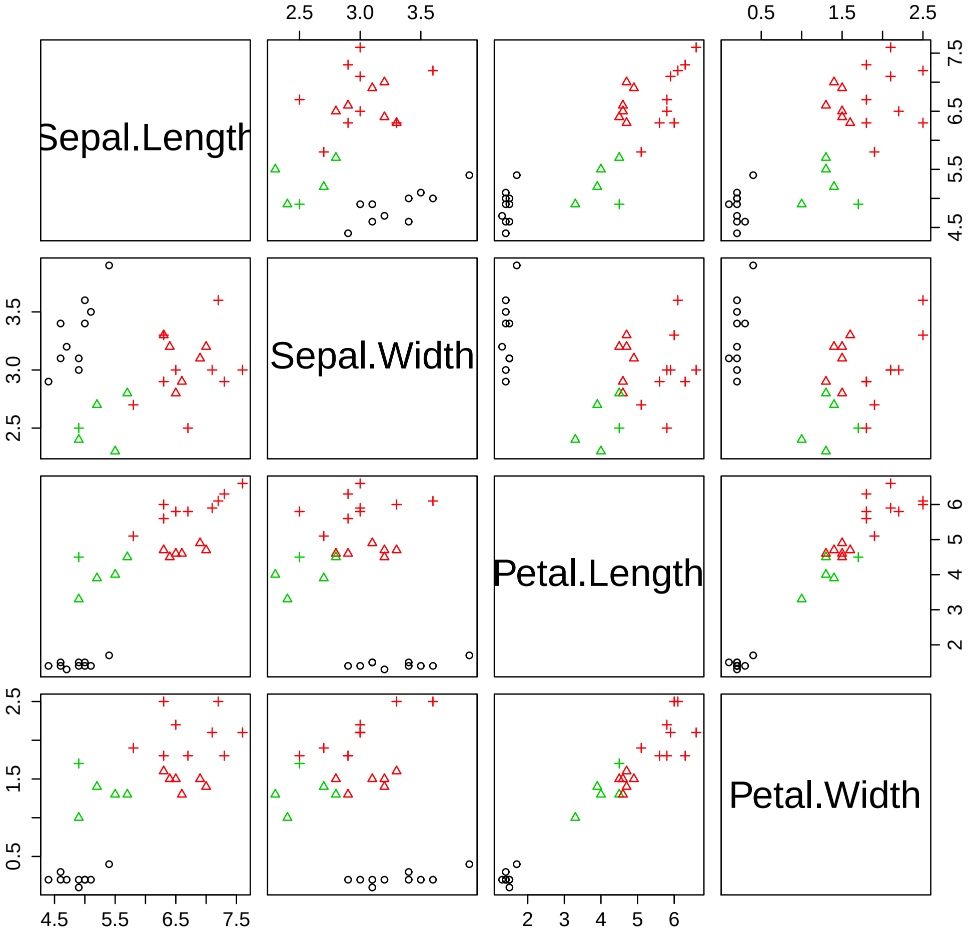
Divisive Clustering - Iris Data

R Syntax for KMeans Clustering into Three Groups

```
### Edited output:
Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1        4.860       3.310     1.450000        0.22
2        6.875       3.025     6.012500        2.10
3        5.975       2.825     4.441667        1.45

Clustering vector:
  1   2   3   4   5   6   7   8   9  10
  1   1   1   1   1   1   1   1   1   1   ### All setosa in Group 1

51  52  53  54  55  56  57  58  59  60
3   3   3   3   3   3   3   3   3   3     ### All versicolor in Group 3

101 102 103 104 105 106 107 108 109 110
  2   3   2   2   2   2   3   2   2   2   ### 8 virginica in Group 2


### Misclassification rate:  2/30 = 6.7 pct
```

Divisive Clustering - Iris Data

Scatterplot Matrix Showing KMeans 3-Group Clusters



```
plot(irispart[, c(1:4)], col=kcluster3$cluster, pch=unclass(irispart[,5]))
```

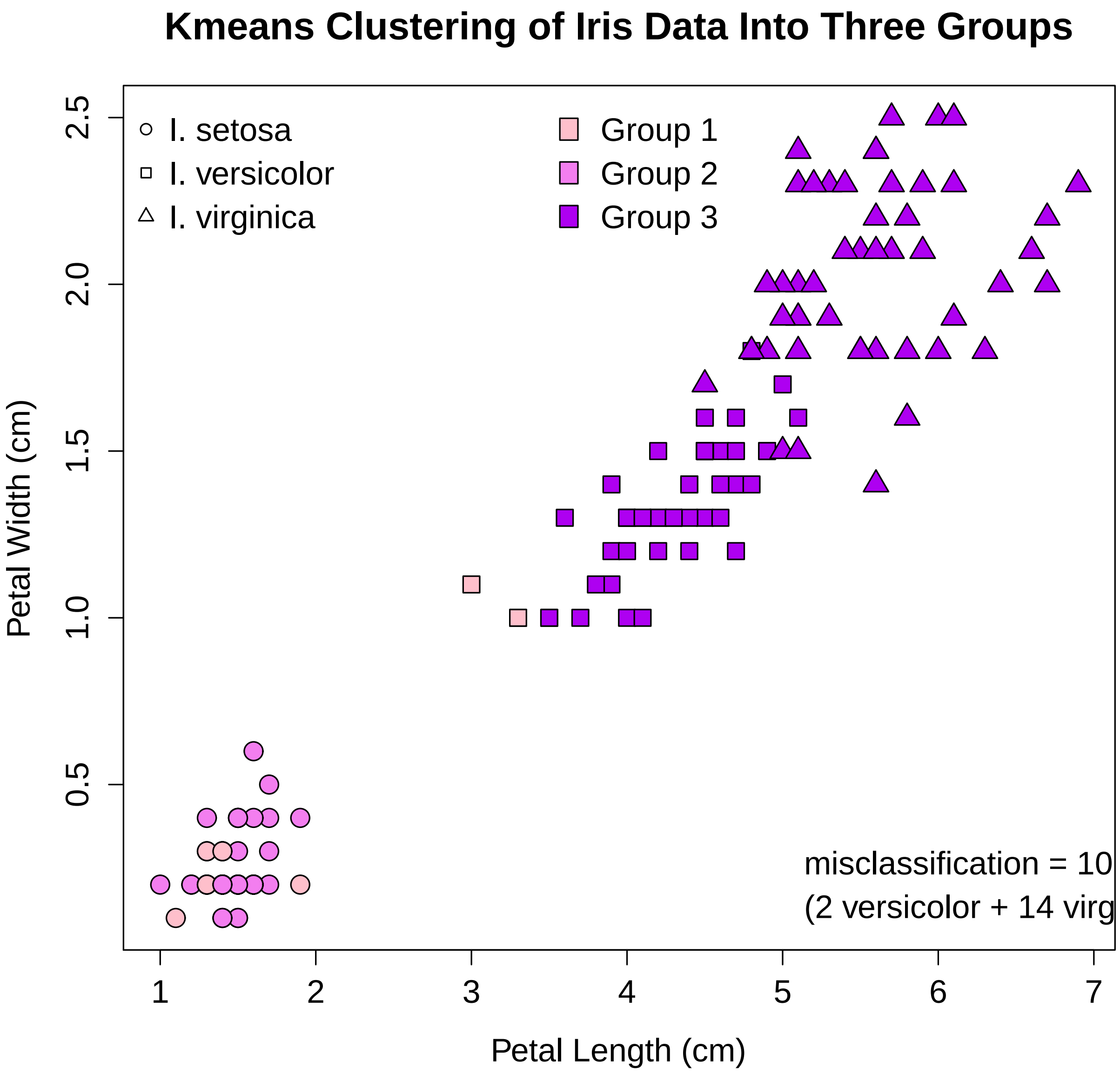
Divisive Clustering - Iris Data

Plotting Results Using Best Two Variables; Adding Association Analysis

```
data(iris); attach(iris)
kcluster3.all <- kmeans(iris[, c(1:4)], 3)
table(Species, kcluster3.all$cluster)
Species       1  2  3
  setosa     50  0  0
  versicolor  0  2 48
  virginica   0 36 14

chisq.test(Species, kcluster3.all$cluster)
 Pearson’s Chi-squared test
data:  Species and kcluster3.all$cluster
X-squared = 223.5993, df = 4, p-value < 2.2e-16

plot(Petal.Length, Petal.Width,
     pch=c(21, 22, 24)[unclass(Species)],
     cex=1.7, xlab="Petal Length (cm)", ylab="Petal Width (cm)",
     main="Kmeans Clustering of Iris Data Into Three Groups",
     bg=c("pink", "violet", "purple")[kcluster3.all$cluster])
legend(x="topleft", c("I. setosa", "I. versicolor", "I. virginica"),
       pch=c(21, 22, 24), bty="n")
legend(x="top", c("Group 1", "Group 2", "Group 3"),
       fill=c("pink", "violet", "purple"), bty="n")
legend(x="bottomright", c("misclassification = 10.7 pct",
       "(2 versicolor + 14 virginica)"), bty="n")
```

Divisive Clustering - Iris Data

Plotting Results Using Best Two Variables; Adding Association Analysis



Advanced Topics - Clustering on Principal Components

Microcosm Test Using Contaminated Sediments

- This example published by Chariton, et al. (2014) following the approach described by Ben-Hur and Guyon (2003)
- Data are from a sediment toxicity test to determine the effects of zero/low/high concentrations of triclosan (antibiotic/antifungal compound) on sediment biota
- Sediment biota were identified using pres/abs molecular markers that identified >850 *different* sediment organisms
- The biota were listed by *operational taxonomic units* (OTUs) rather than genus and species (OTUs)

Clustering on Principal Components

Preliminary Data Decisions

- The source file contained presence/absence data for 858 OTUs from three treatments (control, low, high) with six replicates per treatment
- Nine OTUs were excluded because they had identical values for all samples (variance = zero)
- Final data set contained 18 rows and 849 variables (OTUs)
- The data were analyzed using scaled `prcomp`
- PCA truncated at 18 components (residual variance <3.3e-15)

Clustering on Principal Components

Step 1: Creating New Variables from Component Scores

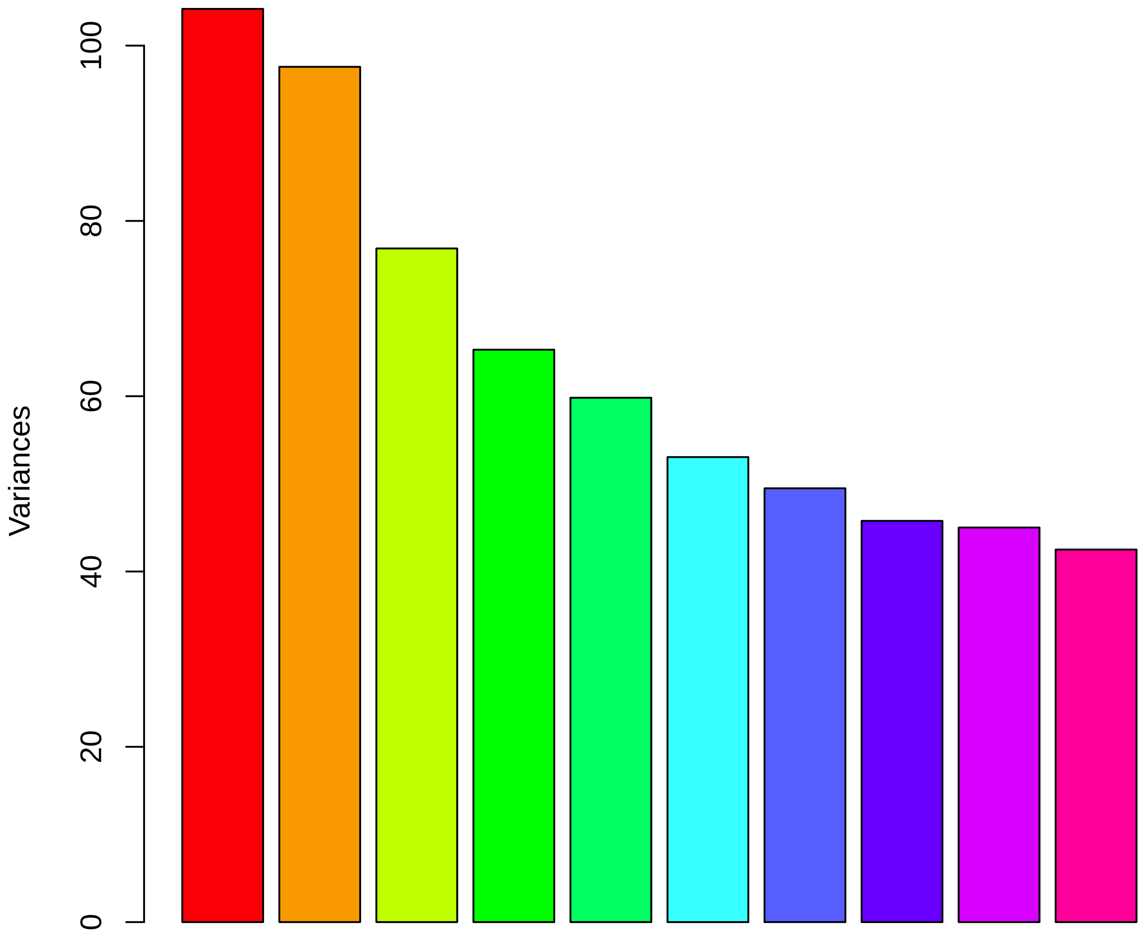
```
#### create the PCA using OTUs (col 1-2 = treatment/replicate)
alldata <- read.csv("alldataOTU.csv", T); attach(alldata)
alldataPCA <- prcomp(alldata[, c(3:851)], scale=T)
summary(alldataPCA)
```

```
Importance of components:
                          PC1   PC2     PC3     PC4    PC5     PC6     PC7
Standard deviation     10.221 9.880 8.77070 8.08297 7.7312 7.28034 7.03630
Proportion of Variance  0.123 0.115 0.09061 0.07695 0.0704 0.06243 0.05832
Cumulative Proportion   0.123 0.238 0.32863 0.40559 0.4760 0.53842 0.59673
                           PC8     PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     6.75020 6.71040 6.52041 6.3837 6.27286 5.86208 5.52219
Proportion of Variance 0.05367 0.05304 0.05008 0.0480 0.04635 0.04048 0.03592
Cumulative Proportion  0.65040 0.70344 0.75352 0.8015 0.84786 0.88834 0.92426
                          PC15    PC16    PC17      PC18
Standard deviation     5.05834 4.56602 4.22722 3.321e-15
Proportion of Variance 0.03014 0.02456 0.02105 0.000e+00
Cumulative Proportion  0.95440 0.97895 1.00000 1.000e+00
```

```
#### write the scores to a new data set:
PCA.scores <- data.frame(alldata$treatment, alldata$replicate, round(alldataPCA$x, 3))
write.table(PCA.scores, "alldataPCA.csv", quote=F, row.names=F, col.names=T, sep=",")
```

Clustering on Principal Components

Variance Plot for First 10 Components



```
plot(alldataPCA, col=rainbow(10), main=" ")
```

Clustering on Principal Components

Comparison of Original and New Data Sets

Original Sediment Microcosm Data (18 rows, 851 columns)

Treatment	Replicate	OTU 1	. . .	OTU 849
control	1	0 or 1		0 or 1
control	2	0 or 1		0 or 1
(etc.)	(etc.)	(etc.)		(etc.)
high	5	0 or 1		0 or 1
high	6	0 or 1		0 or 1

PCA Data - `alldataPCA$x` **(18 rows, 20 columns)** -

Treatment	Replicate	PC 1	. . .	PC 18
control	1	-4.97		$< \pm 0.01$
control	2	-15.53		$< \pm 0.01$
(etc.)	(etc.)	(etc.)		(etc.)
high	5	13.81		$< \pm 0.01$
high	6	13.10		$< \pm 0.01$

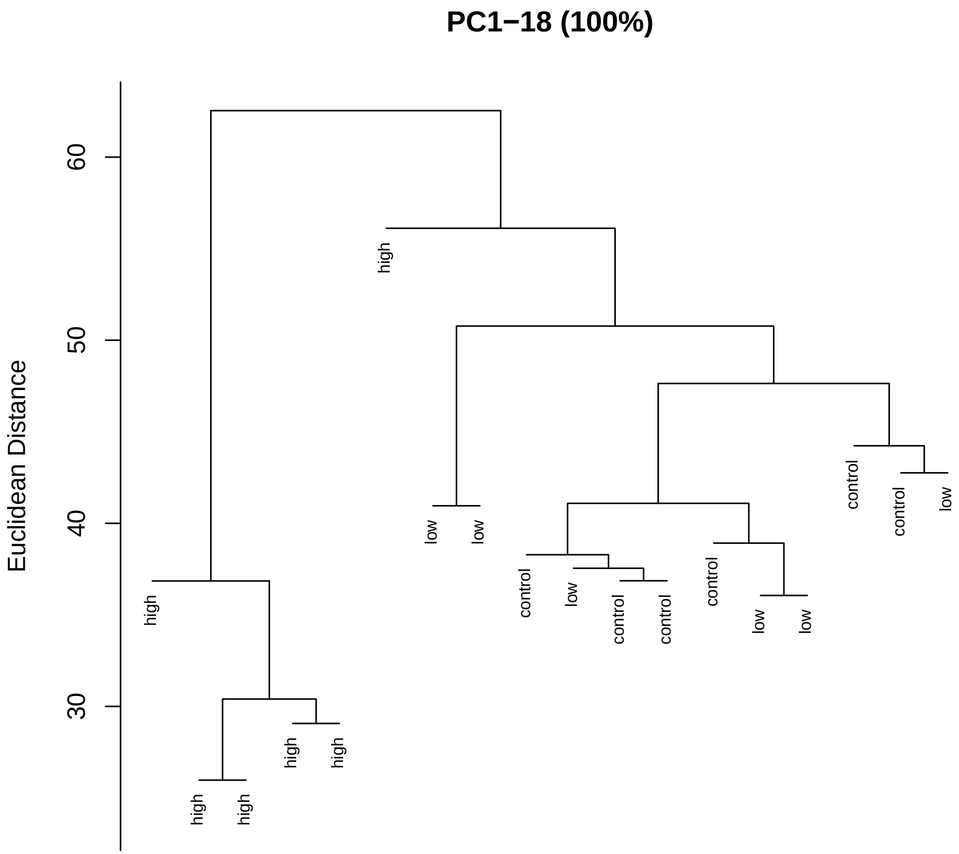
Clustering on Principal Components

Step 2: Clustering on the Component Scores

- The next process is based on the fact that the scaled, centered PCA creates a multivariate correlation matrix, with the "best" correlations contained in the first component
- Each successive component containing a smaller fraction of "good" correlation
- We want to cluster using the smallest subset of components that will produce stable clusters . . . this is a significant departure from traditional ordination
- The next figure shows initial euclidean/wards hierarchical clustering using all 18 principal components as a starting point

Clustering on Principal Components

Dendrogram Results using 18 Components



```
newdata <- read.csv("alldataPCA.csv", T); attach(newdata)
distances <- dist(newdata[, c(3:20)], method="euclidean")
eward <- hclust(distances, method="ward")
plot(eward, labels=treatment, hang=0, cex=0.65, xlab=" ", sub=" ",
      main="PC1-18 (100)", ylab="Euclidean Distance")
```

Clustering on Principal Components

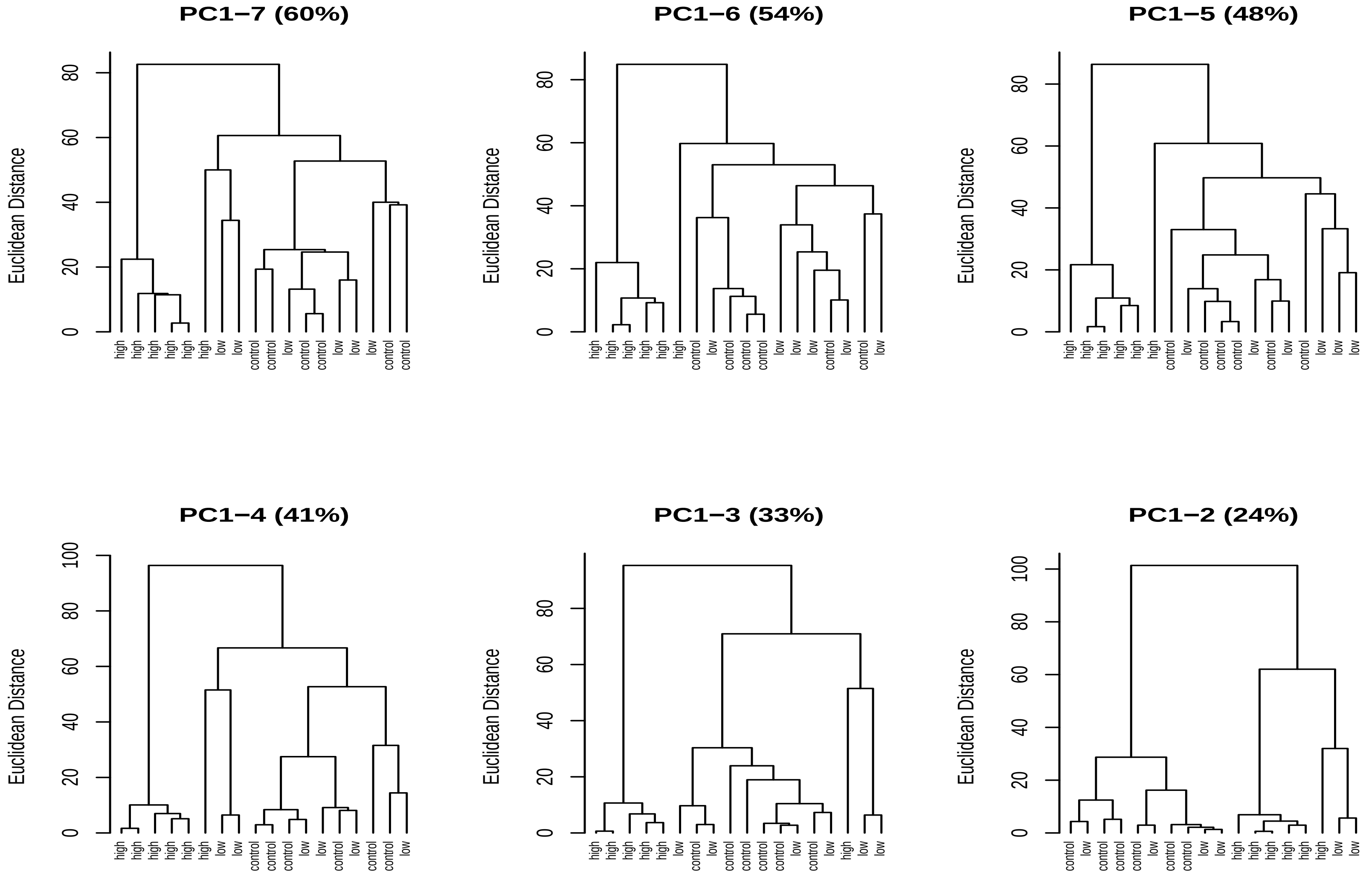
Step 3: Identifying Stable Clusters

- Selecting the fewest components for stable clustering is actually a complex process (see Ben-Hur and Guyon, 2003)
- Preliminary evaluation of the 18-component clusters reveal that there are only two *treatment* responses (high vs. control+low)
- Using `cuttree` and `table`, we can look at the number of misclassifications between the cluster groups and treatment, with misclassification defined as samples that don't match "high" or "control+low
- Cycling through all dendrograms, (PC1–PC18, PC1–PC17, PC1–PC16, etc), each results in 1 misclassification until the final option (PC1–PC2), which results in 2 misclassifications

  You only need PC1–PC3 to produce stable clusters

Clustering on Principal Components

Dendrogram Results using First Seven Principal Components



PC1−7 (60%)

PC1−6 (54%)

PC1−5 (48%)

PC1−4 (41%)

PC1−3 (33%)

PC1−2 (24%)

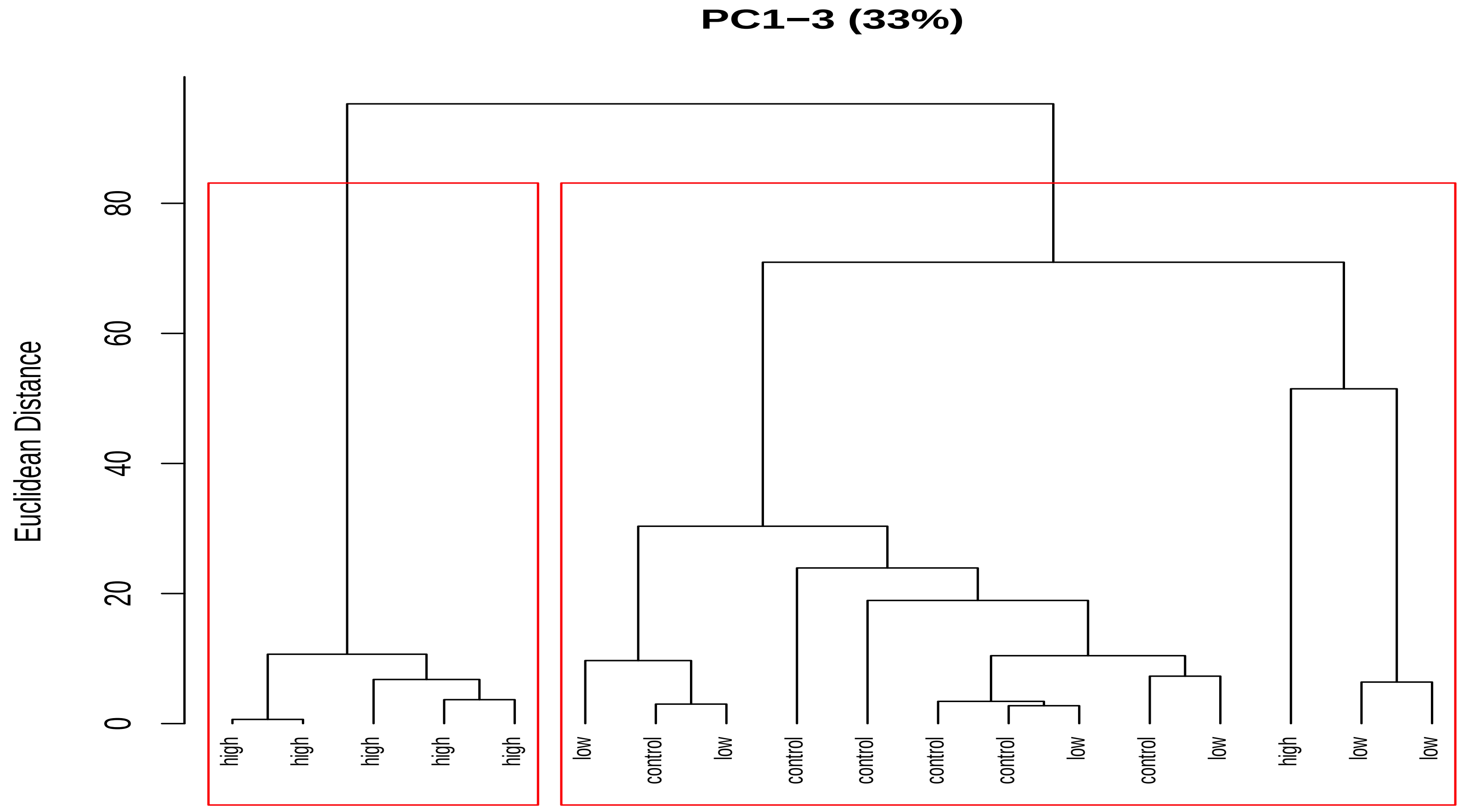
Clustering on Principal Components

Step 4: Refining Cluster Membership and Checking Significance

- After selecting the minimum number of components for clustering, we should review cluster membership
- The figure on page 41 shows how two PCA cluster groups match "high" and "low+control" treatments, with one misclassification
- But the figure on page 42 reveals that you could also describe the data using three clusters
  - Two of the clusters show treatment effects (high or low+control)
  - The third cluster contains three outlier samples
- Choosing which way to display the results depends on your overall goals, but it is usually desirable to discuss outliers separately from the treatment effect

Clustering on Principal Components

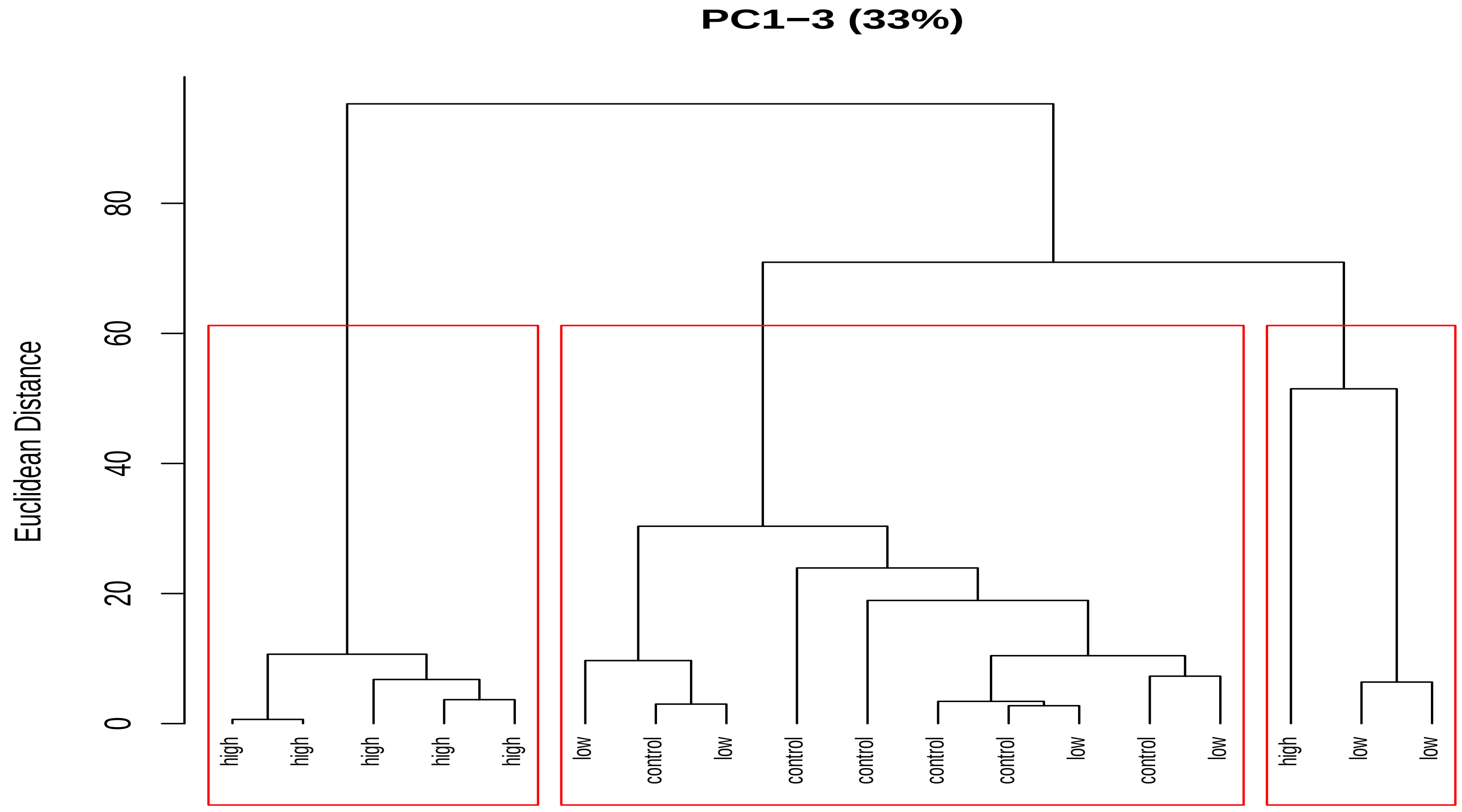
Two-Cluster Dendrogram



```
HCgroups <- cutree(eward, 2); chisq.test(HCgroups, treatment)
X-squared = 13.8462, df = 2, p-value = 0.0009848
```

Clustering on Principal Components

Three-Cluster Dendrogram



```
HCgroups <- cutree(eward, 3); chisq.test(HCgroups, treatment)
X-squared = 17.6, df = 4, p-value = 0.001477
```

Clustering on Principal Components

Step 5: Interpreting the Results

- Using this approach, the data revealed two treatment responses (not three) and a subgroup of outliers from different treatment groups
- To finish the evaluation, we can examine how the source data influenced the principal components
- With continuous data (e.g., water quality, algae counts), you could use summary statistics (e.g., minimum, median, maximum) for each cluster group
- Summary statistics are not helpful for presence/absence data, so we examined the top 10 negative and positive OTU scores for the PC1–PC3 (`alldataPCA$rotation`; table on page 44)

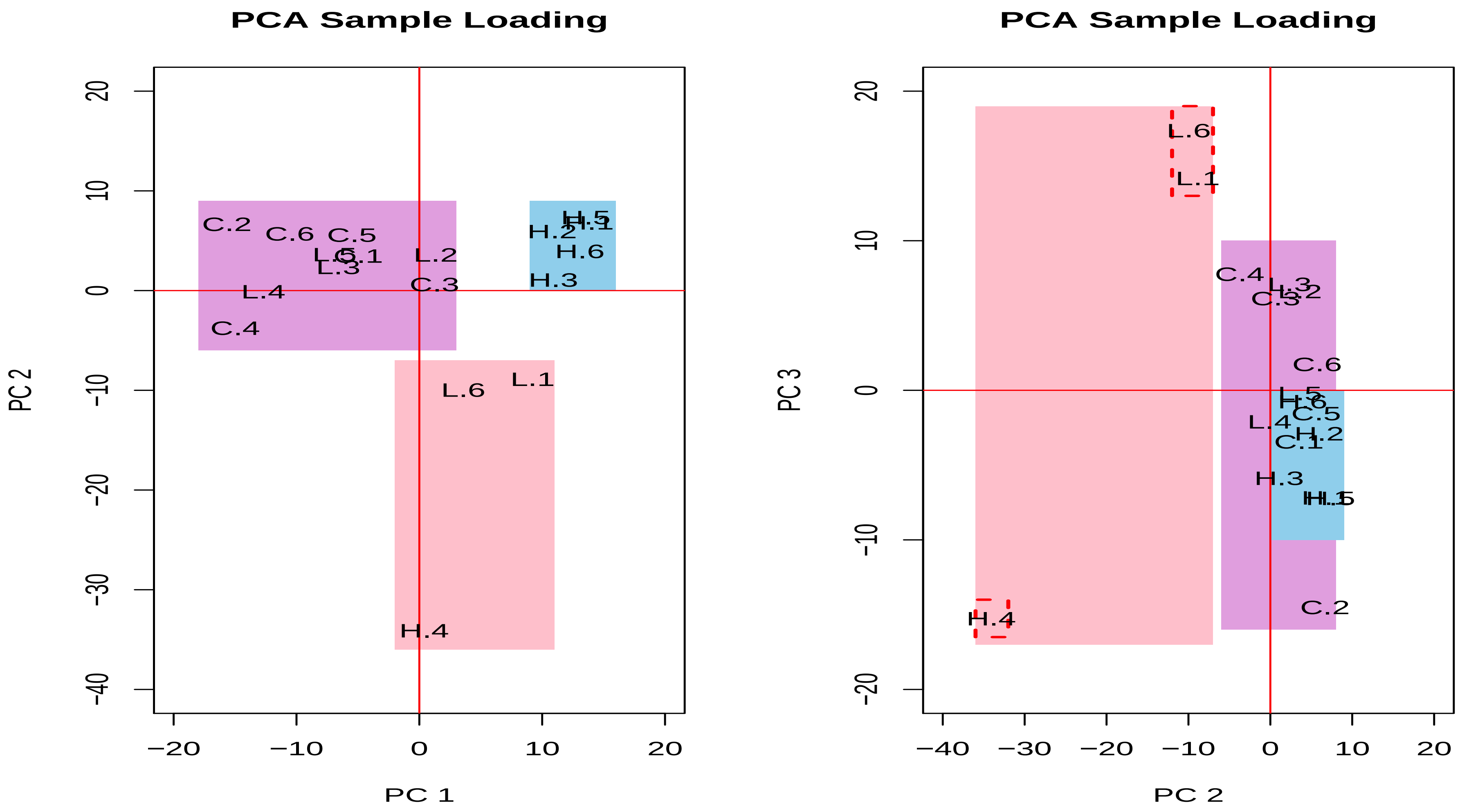
Best Negative Variable Scores in PC1-PC3

OTU	PC1	OTU	PC2	OTU	PC3
M.5324	-0.085	M.521	-0.087	M.33548	-0.082
M.18385	-0.083	F.671	-0.087	M.29634	-0.076
M.17875	-0.083	uk.euk.3152	-0.087	uk.euk.29604	-0.074
M.34715	-0.083	S.3978	-0.087	M.10729	-0.074
M.25065	-0.082	S.7008	-0.087	M.4300	-0.073
M.28527	-0.082	M.7687	-0.087	R.848	-0.071
uk.euk.6428	-0.082	S.9894	-0.087	uk.euk.6422	-0.071
M.1091	-0.081	S.10341	-0.087	S.947	-0.071
M.15718	-0.080	S.13318	-0.087	uk.euk.11428	-0.071
M.25537	-0.079	S.15201	-0.087	uk.euk.4378	-0.071

Best Positive Variable Scores in PC1-PC3

OTU	PC1	OTU	PC2	OTU	PC3
R.1633	0.056	F.35789	0.037	M.20011	0.075
M.37505	0.056	uk.euk.28316	0.037	R.27994	0.075
R.28351	0.056	M.588	0.038	A.25541	0.075
M.37021	0.057	M.25299	0.039	M.29451	0.075
M.37060	0.060	uk.euk.35868	0.040	M.26907	0.075
S.3563	0.065	M.10534	0.040	M.35148	0.076
R.9197	0.066	M.15282	0.041	M.35761	0.080
Am.5381	0.068	S.37132	0.044	uk.euk.8723	0.081
A.11234	0.073	M.4361	0.048	uk.euk.4435	0.084
R.30870	0.077	M.5776	0.052	uk.euk.5340	0.088

Plotting the Samples by PCA Scores



PC1 separated the two treatment groups; PC2 and PC3 were useful for separating the outliers

Supplemental References

- Crawley, Michael J. 2013. The R Book. John Wiley & Sons. ISBN 978-0-470-97392-9.
- Everitt, Brian S. 2011. Cluster Analysis, 5th Edition. Wiley, ISBN 978-0-470-74991-3.
- Lander, Jared P. 2014. R for Everyone, Advanced Analytics and Graphics. Addison Wesley Data & Analytics Series, ISBN 978-0-321-88803-7.
- Pielou, Evelyn C. 1984. The Interpretation of Ecological Data: A Primer on Classification and Ordination. Wiley. 978-0-471-88950-2.
- Teetor, Paul. 2011. The R Cookbook. O'Reilly Publishers. ISBN 978-0-596-880915-7

Citations for PCA Clustering

- Ben-Hur, A. and I. Guyon. 2003. Detecting stable clusters using principal component analysis in methods in molecular biology. In Brownstein, M. J. and A. Kohodursky, eds, *Functional Genomics: Methods and Protocols.*, Humana Press, Totowa, NJ, pp 159–182.
- Chariton, A. A., K. T. Ho, D. Proestou, H. Bik, S. L. Simpson, L. M. Portis, M. G. Cantwell, J. G. Baguley, R. M. Burgess, M. M. Pelletier, M. Perron, C. Gunsch, and R. A. Matthews. 2014. A molecular-based approach for examining responses of eukaryotes in microcosms to contaminant-spiked estuarine sediments. *Environmental Toxicology and Chemistry* 33:359–369.