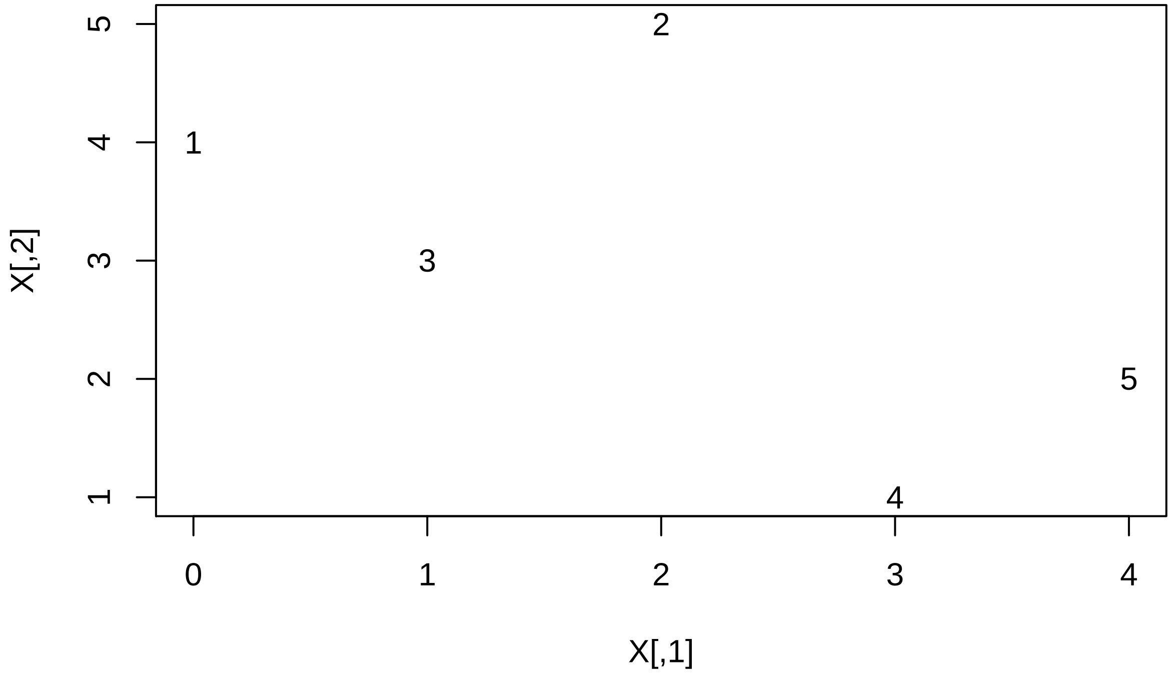
Dissimilarity

```r
set.seed(1)
X <- cbind(c(0, 2, 1, 3, 4), c(4, 5, 3, 1, 2))
plot(X, pch = as.character(1:5))
```



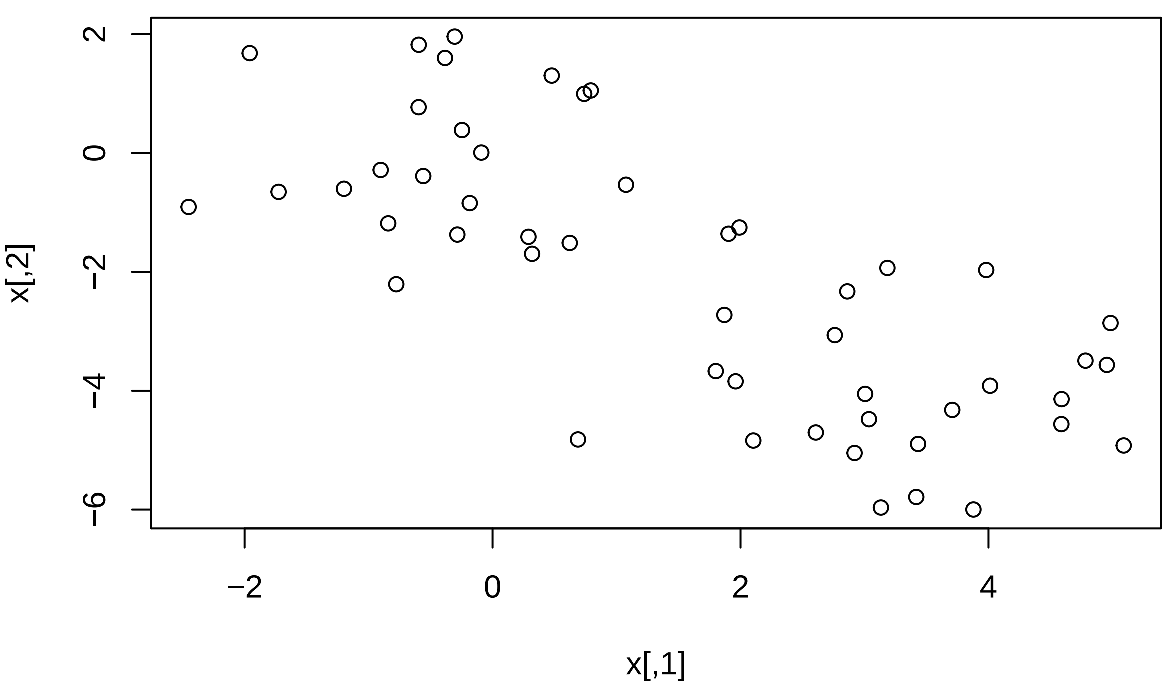
```r
d <- dist(X)
as.matrix(d)
```

```
##          1        2        3        4        5
## 1 0.000000 2.236068 1.414214 4.242641 4.472136
## 2 2.236068 0.000000 2.236068 4.123106 3.605551
## 3 1.414214 2.236068 0.000000 2.828427 3.162278
## 4 4.242641 4.123106 2.828427 0.000000 1.414214
## 5 4.472136 3.605551 3.162278 1.414214 0.000000
```

K-Mean

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

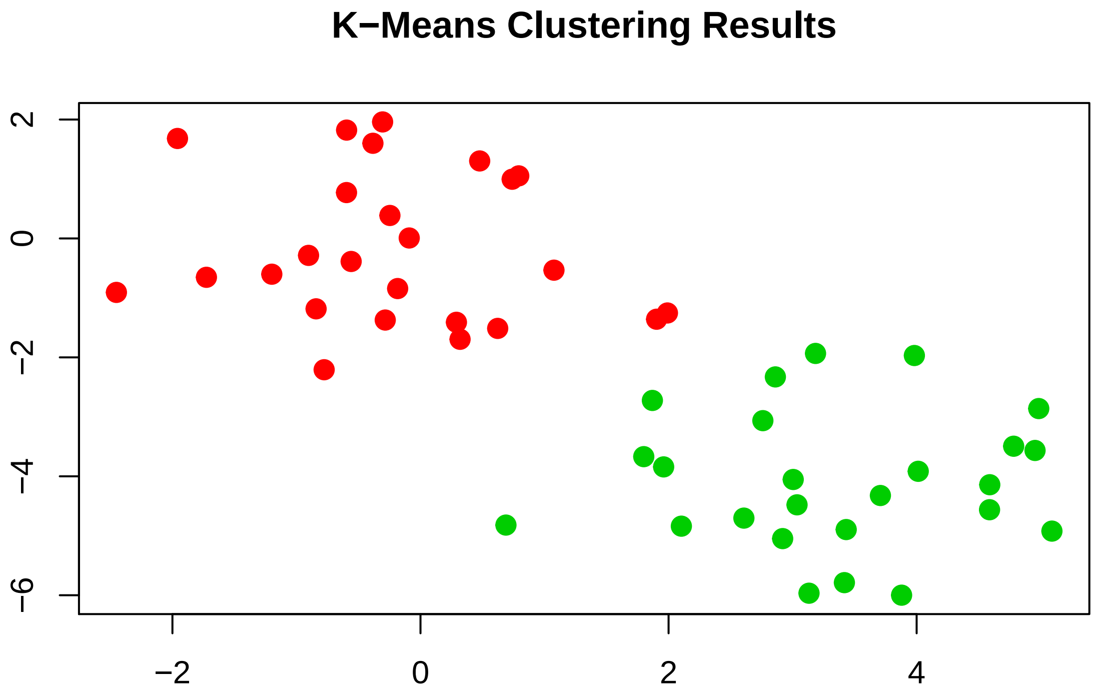
```
x[1:25, 2] <- x[1:25, 2] - 4
plot(x)
```



```
K <- 2
km_out <- kmeans(x, K, nstart = 5)
km_out$cluster
```

```
##  [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 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

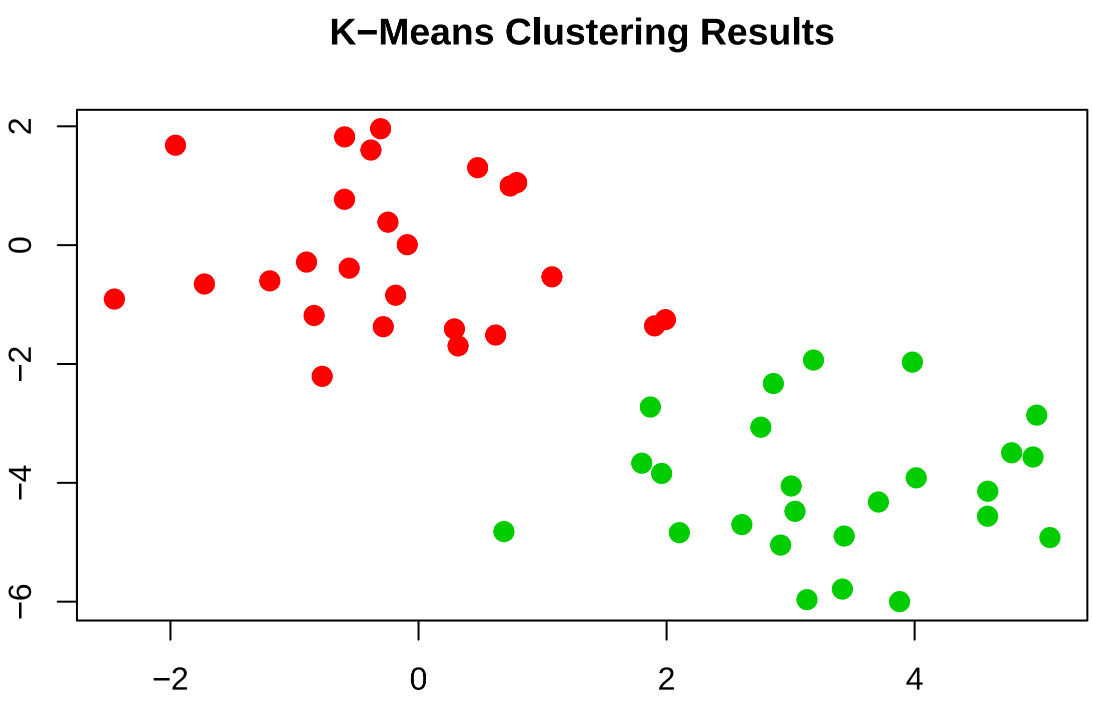
```
plot(x, col = (km_out$cluster + 1), main = "K-Means Clustering Results", xlab = "", ylab = "", pch = 20
```



```r
set.seed(4)
km_out <- kmeans(x, K, nstart = 20)
km_out
```

```
## K-means clustering with 2 clusters of sizes 25, 25
##
## Cluster means:
##         [,1]       [,2]
## 1 -0.1956978 -0.1848774
## 2  3.3339737 -4.0761910
##
## Clustering vector:
##  [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 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 65.40068 63.20595
##  (between_SS / total_SS =  72.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
plot(x, col = (km_out$cluster + 1), main = "K-Means Clustering Results", xlab = "", ylab = "", pch = 20
```



```r
set.seed(3)
km_out <- kmeans(x, K, nstart = 1)
km_out$tot.withinss
```

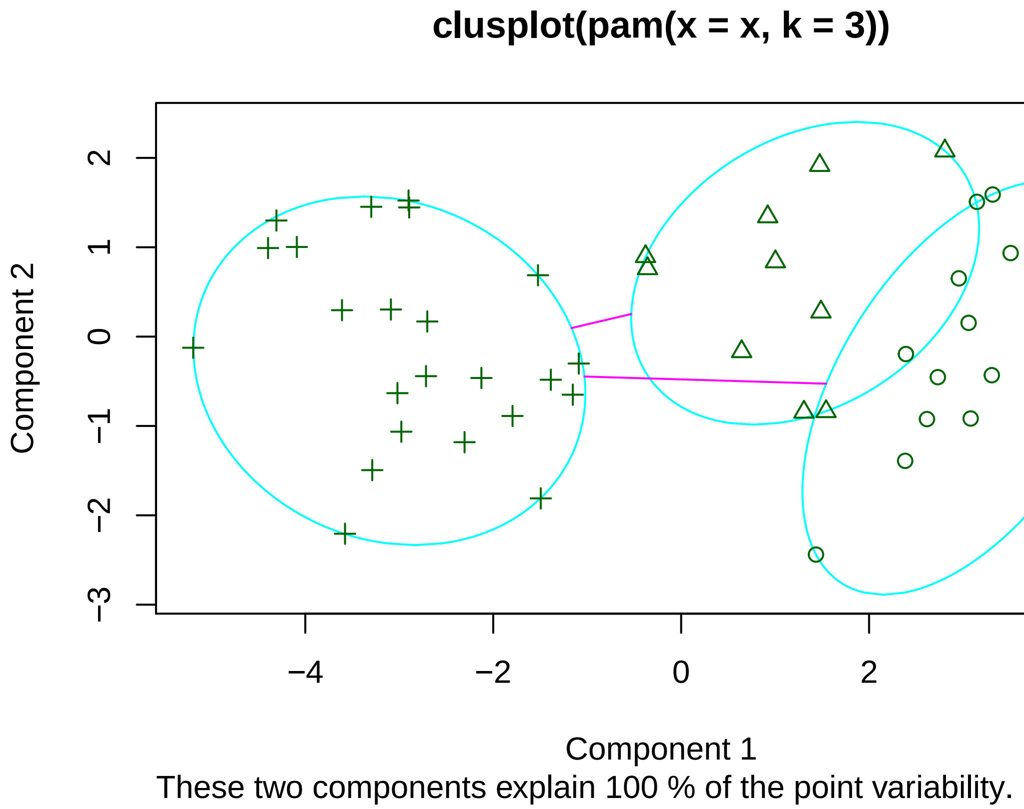
```
## [1] 128.6066
```

```r
K <- 4
n <- nrow(x)
km_out <- kmeans(x, K, nstart = 20)
# CH index
(km_out$betweenss / (K - 1)) / (km_out$tot.withinss / (n - K))
```

```
## [1] 88.77696
```

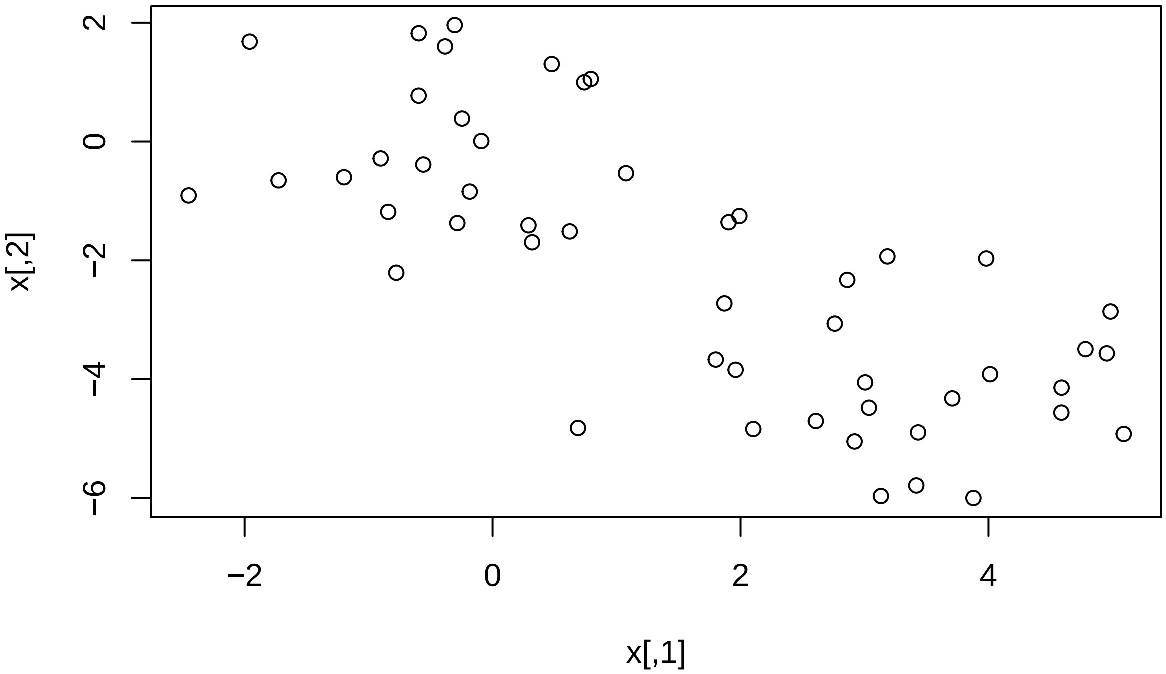
K-Medoids

```r
library(cluster)
pam.fit <- pam(x, 3)
clusplot(pam.fit)
```



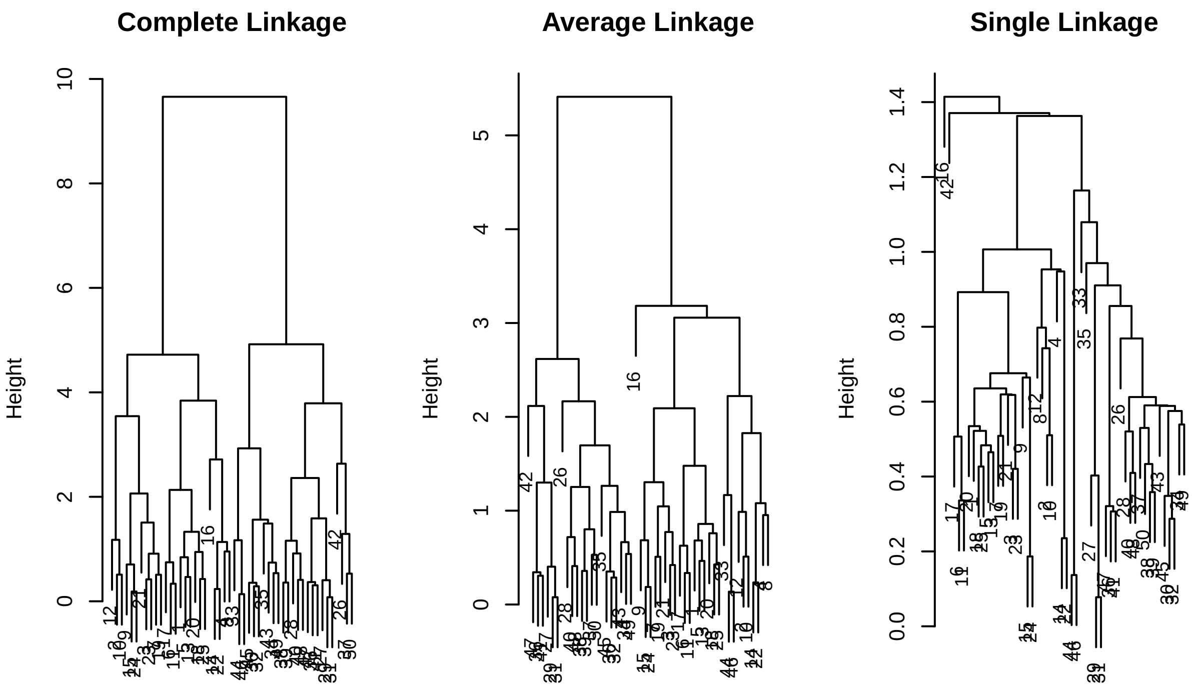
Hierarchical clustering

```
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
plot(x)
```



```
hc_complete <- hclust(dist(x), method = "complete")
hc_average <- hclust(dist(x), method = "average")
hc_single <- hclust(dist(x), method = "single")
```

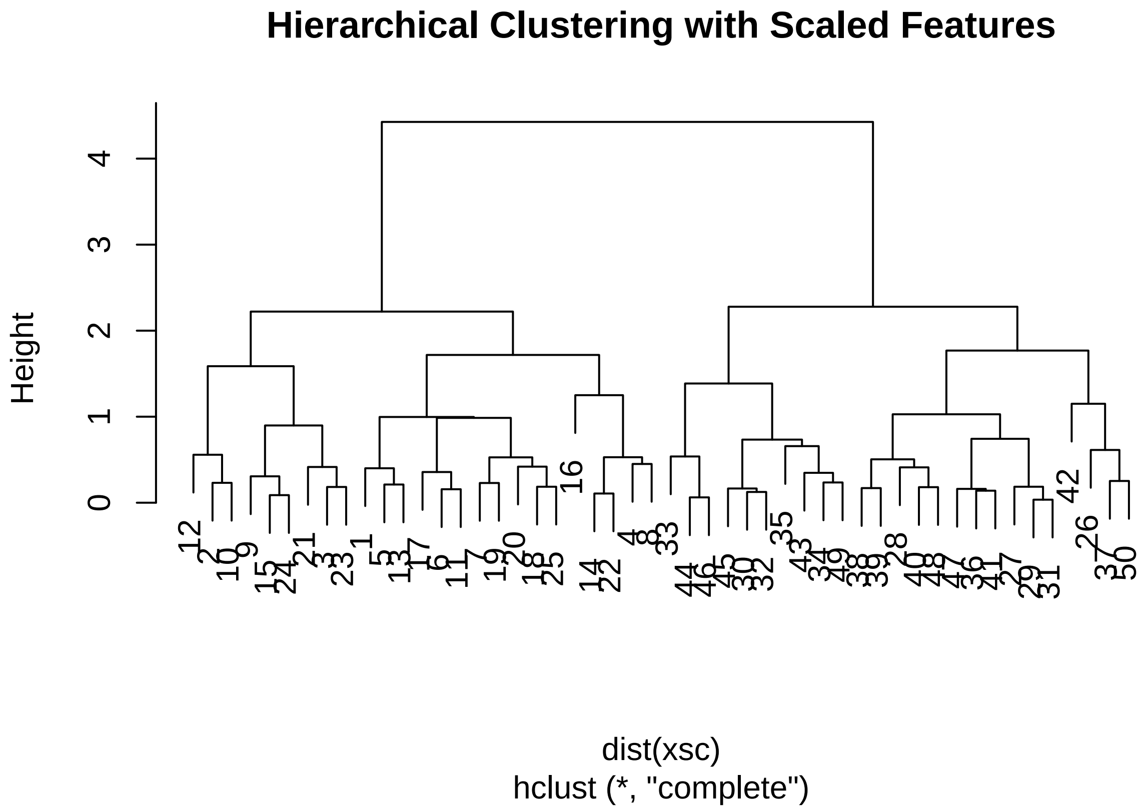
```
par(mfrow = c(1, 3))
plot(hc_complete, main = "Complete Linkage", xlab = "", sub = "", cex = .9)
plot(hc_average, main = "Average Linkage", xlab = "", sub = "", cex = .9)
plot(hc_single, main = "Single Linkage", xlab = "", sub = "", cex = .9)
```



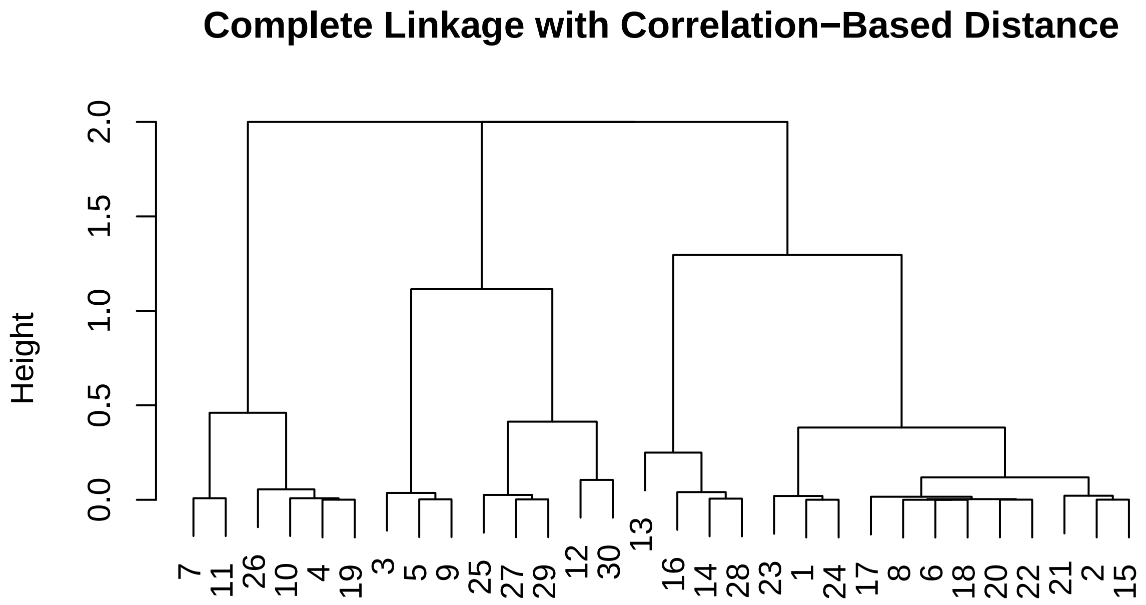
```r
cutree(hc_complete, 3)
```

```
##  [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 3 2 3 3 3 3
## [36] 2 2 2 2 2 2 2 3 3 3 3 2 2 3 2
```

```r
xsc <- scale(x)
plot(hclust(dist(xsc), method = "complete"), main = "Hierarchical Clustering with Scaled Features")
```



```
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 =
```



A Text mining example

```
library(stringr)
wiki <- "http://en.wikipedia.org/wiki/"
titles <- c(
    "Integral", "Riemann_integral", "Riemann-Stieltjes_integral", "Derivative",
    "Limit_of_a_sequence", "Edvard_Munch", "Vincent_van_Gogh", "Jan_Matejko",
    "Lev_Tolstoj", "Franz_Kafka", "J._R._R._Tolkien"
)
articles <- character(length(titles))

for (i in 1:length(titles)) {
    articles[i] <- str_flatten(readLines(paste0(wiki, titles[i])), col = " ")
}
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
docs <- Corpus(VectorSource(articles)) %>%
    tm_map(content_transformer(function(z) str_replace(z, "<.+?>", " "))) %>%
    tm_map(content_transformer(function(x) str_replace(x, fixed("\t"), " "))) %>%
    tm_map(PlainTextDocument) %>%
    tm_map(removePunctuation) %>%
    tm_map(stripWhitespace) %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removeWords, stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(function(z)
## str_replace(z, : transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(function(x)
## str_replace(x, : transformation drops documents

## Warning in tm_map.SimpleCorpus(., PlainTextDocument): transformation drops
## documents

## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
docsTDM <- TermDocumentMatrix(docs)
```

```
t(as.matrix(docsTDM))[, sample(10000, 3)]
```

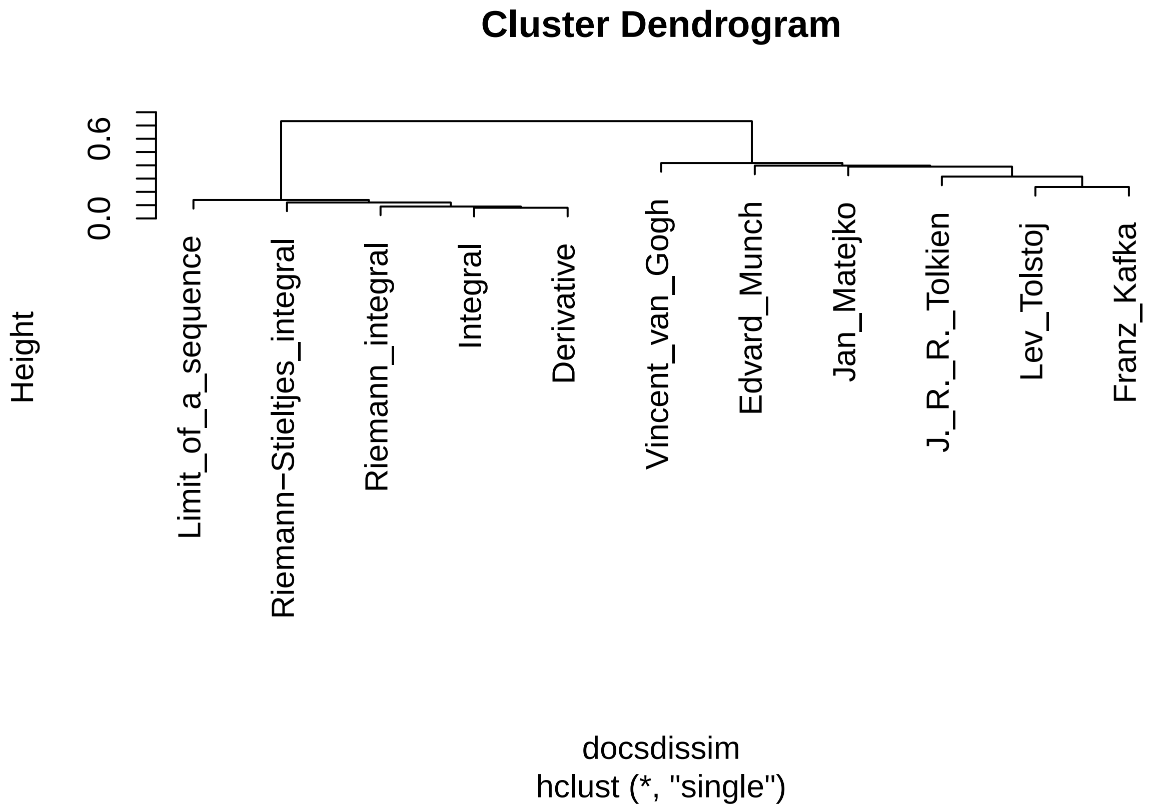
```
##     Terms
## Docs classtoclevel3 tdtr stylepadding03em
##   1               2    4                1
##   2               0    0                1
##   3               0    0                0
##   4               0    0                1
##   5               0    5                0
##   6               0    0                0
##   7              13    0                0
##   8               0   26                0
##   9               0    0                1
##   10              3   25                0
##   11             27    6                0
```

```
library(proxy)
```

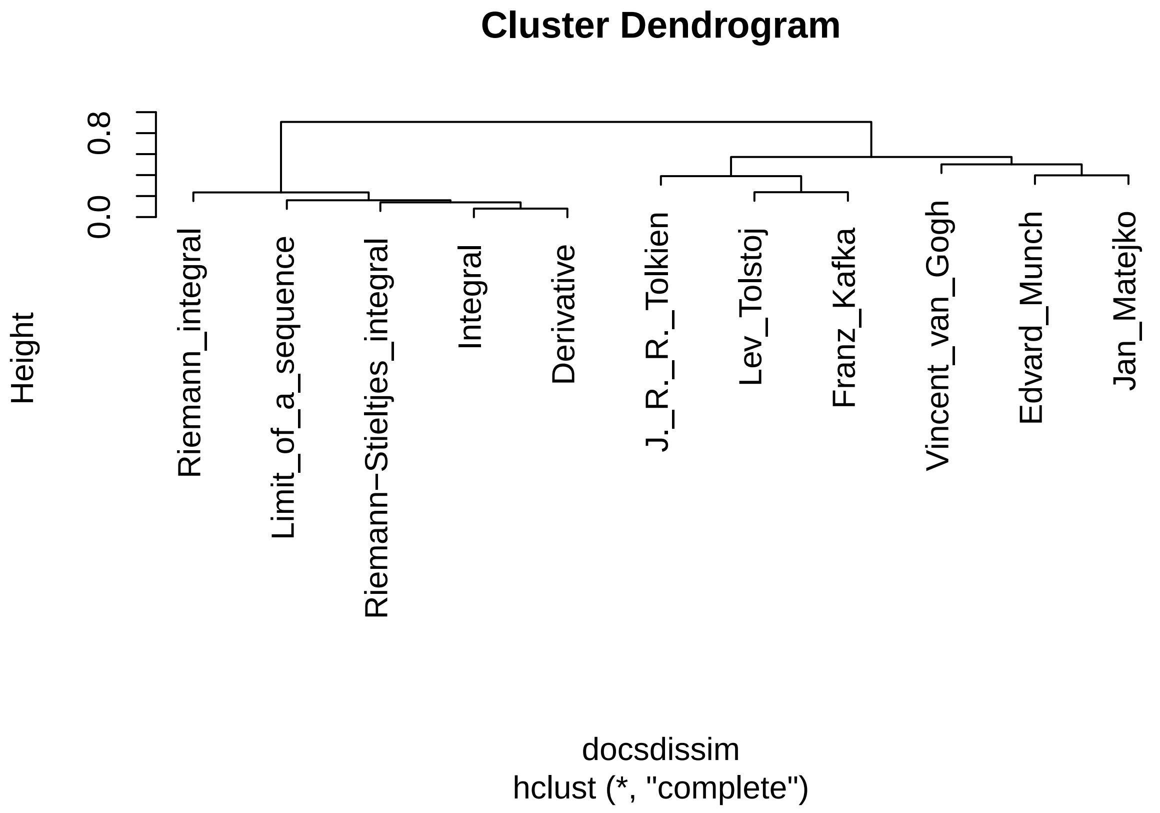
```
##
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix
```

```
docsdissim <- dist(t(as.matrix(docsTDM)), method = "cosine")
```

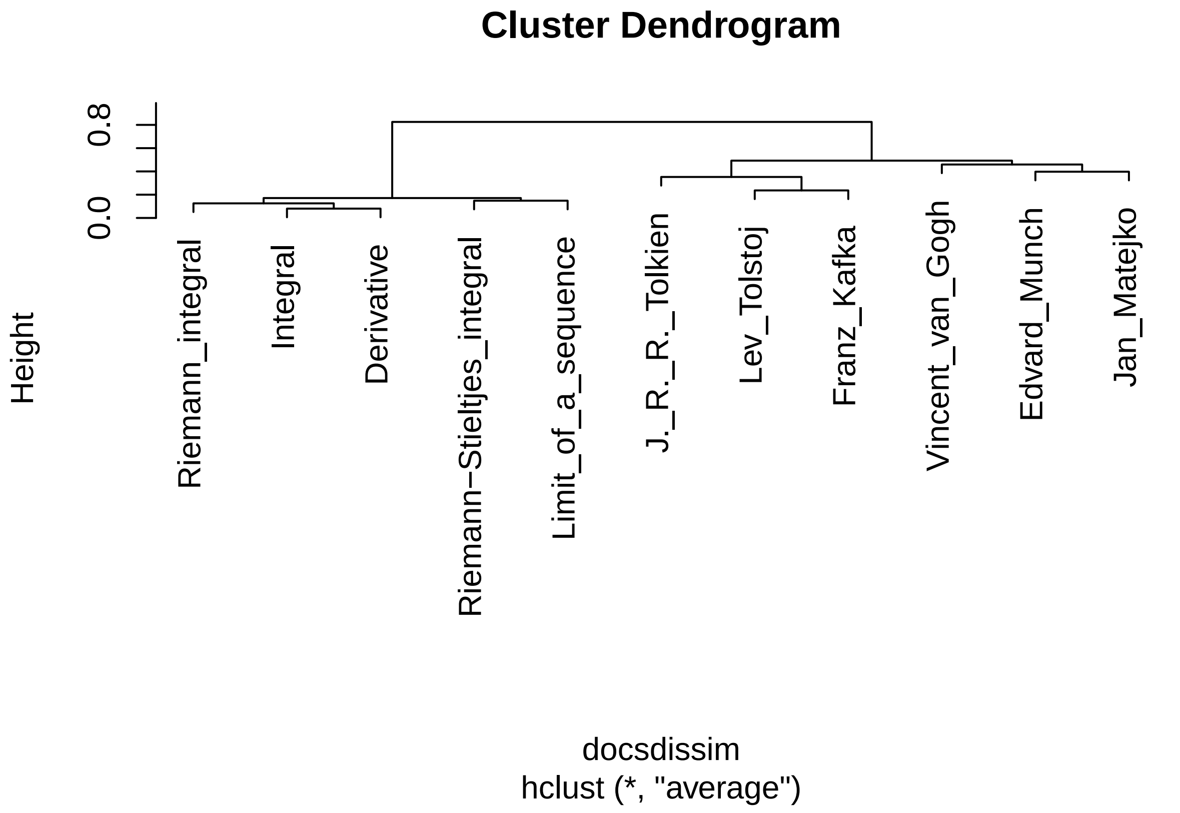
```
h <- hclust(docsdissim, method = "single")
plot(h, labels = titles)
```



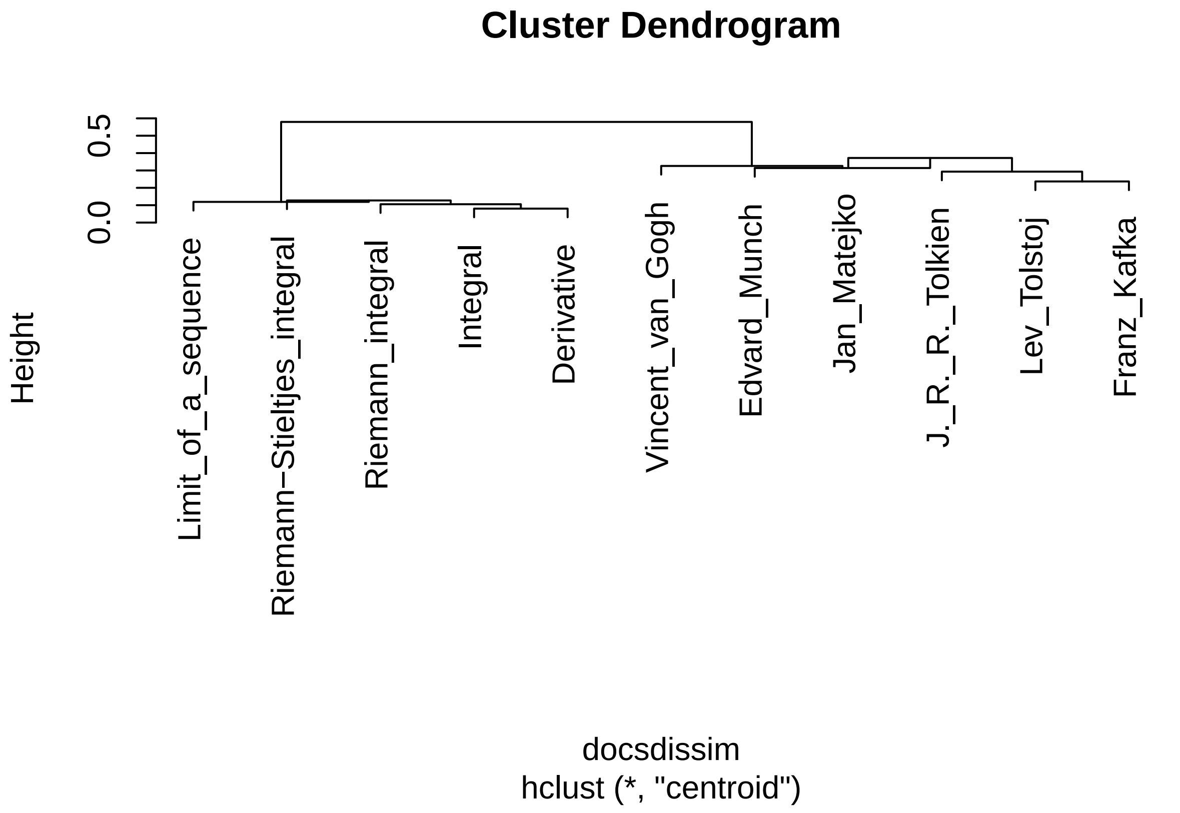
```
h <- hclust(docsdissim, method = "complete")
plot(h, labels = titles)
```



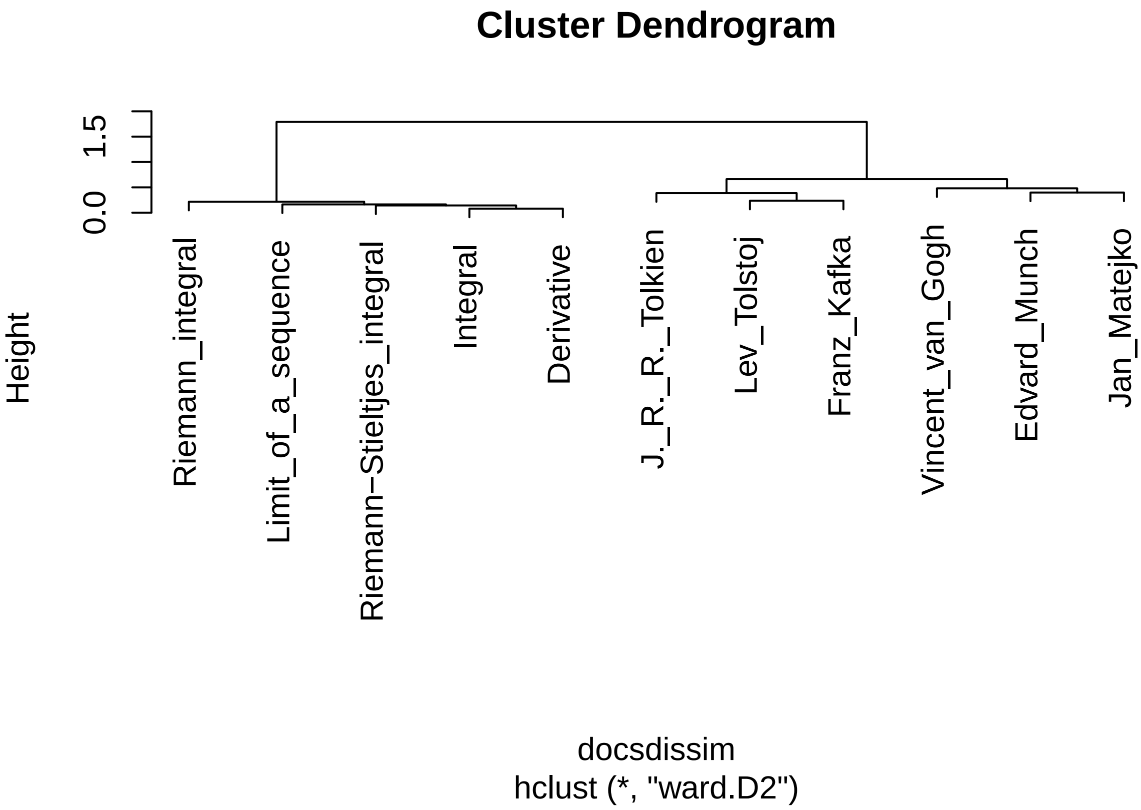
```
h <- hclust(docsdissim, method = "average")
plot(h, labels = titles)
```



```
h <- hclust(docsdissim, method = "centroid")
plot(h, labels = titles)
```



```
h <- hclust(docsdissim, method = "ward.D2")
plot(h, labels = titles)
```


Cluster Dendrogram
Height
0.0
1.5
Riemann_integral
Limit_of_a_sequence
Riemann−Stieltjes_integral
Integral
Derivative
J._R._R._Tolkien
Lev_Tolstoj
Franz_Kafka
Vincent_van_Gogh
Edvard_Munch
Jan_Matejko
docsdissim
hclust (*, "ward.D2")