Module 10: Recommended Exercises

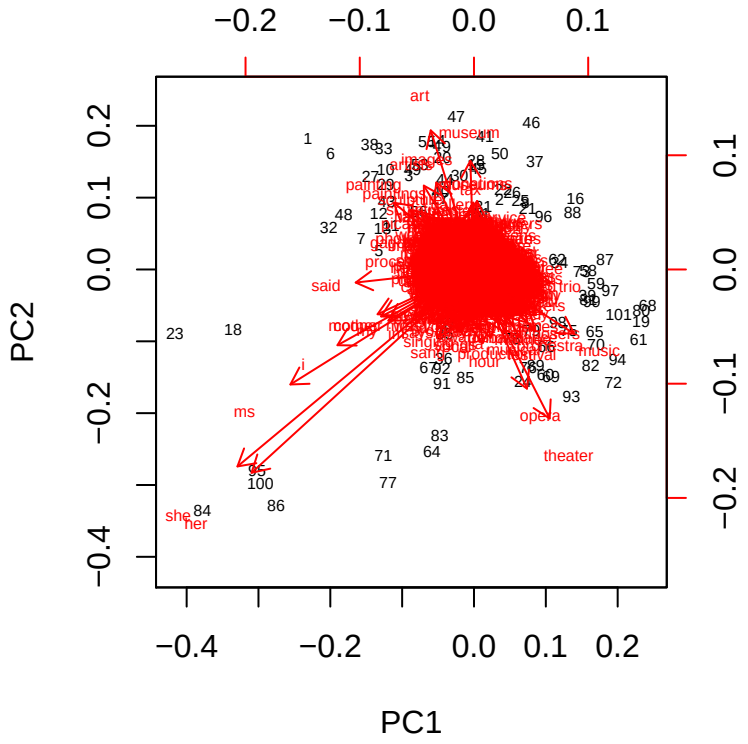
Statistical Learning V2021

alexaoh

22 mai, 2021

Problem 1

```
load("pca-examples.rdata")
nyt.data <- nyt.frame
pr.out <- prcomp(nyt.data[, -1])  # Remove the first column, which is a factor.
biplot(pr.out, scale = 0, cex = 0.5)  # There is a lot of information here!
```



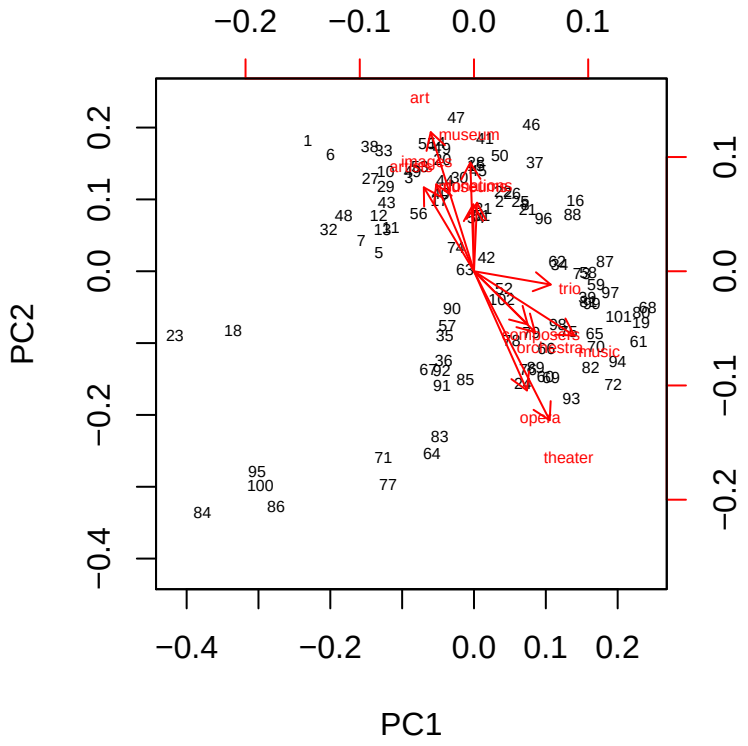

```
# Very hard to interpret. Should select some loadings to focus on.

nyt.loading <- pr.out$rotation[, 1:2]  # Select loadings from PC1 and PC2

# Find the first most important loadings in PC1 and PC2 (with largest
# weights).
```

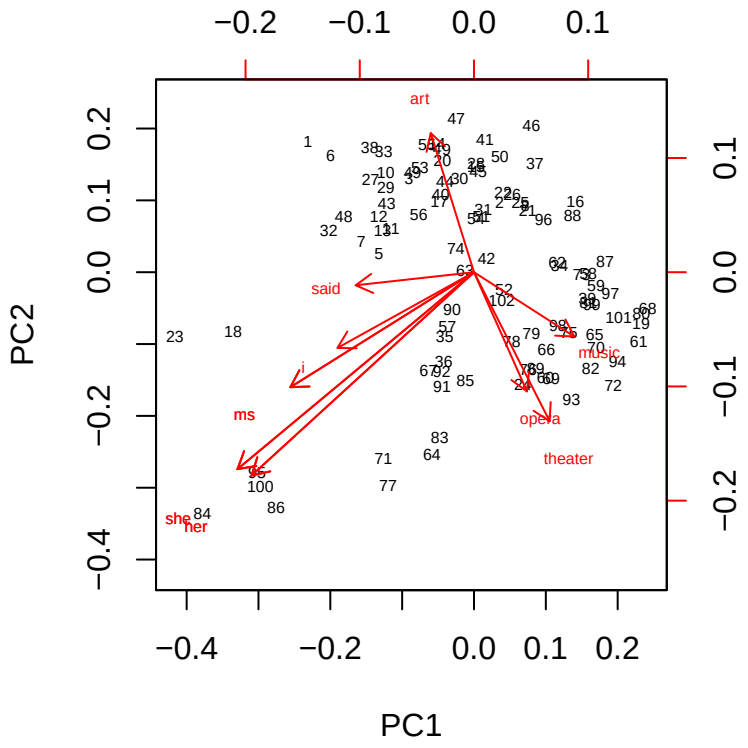
```
informative.loadings <- rbind(head(nyt.loading[order(nyt.loading[, 1],
    decreasing = T), ]), head(nyt.loading[order(nyt.loading[, 2], decreasing = T),
    ]))

biplot(x = pr.out$x[, 1:2], y = informative.loadings, scale = 0, cex = 0.5)
```



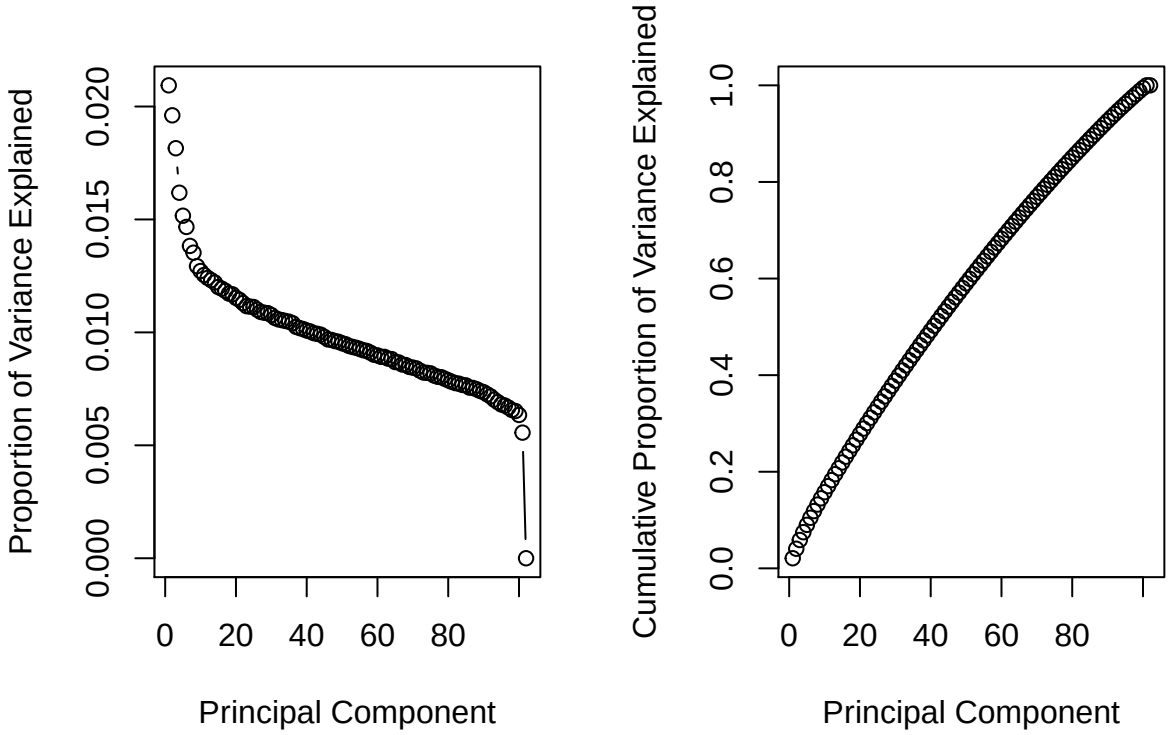

```
# Find the first most important loadings in PC1 and PC2 (with largest
# weights in absolute value).
informative.loadings.abs <- rbind(head(nyt.loading[order(abs(nyt.loading[,
    1]), decreasing = T), ]), head(nyt.loading[order(abs(nyt.loading[,
    2]), decreasing = T), ]))

biplot(x = pr.out$x[, 1:2], y = informative.loadings.abs, scale = 0,
    cex = 0.5)
```




```
par(mfrow = c(1, 2))
# PVE
pr.var <- pr.out$sdev^2
pve <- pr.var/sum(pr.var)
plot(pve, xlab = "Principal Component", ylab = "Proportion of Variance Explained",
    type = "b")

# Cumulative PVE
plot(cumsum(pve), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained",
    type = "b")
```



From these types of plots one can extract information about which principal components one should use in order to explain a given amount of variance in the original data. These are visual tools that can be used, instead of only looking at the data given in the summary from `prcomp`.

Problem 2

Show that Algorithm 10.1 for K-Means Clustering is guaranteed to decrease the value of the objective

$$\min_{C_1,\ldots,C_k} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

at each step.

Problem 3

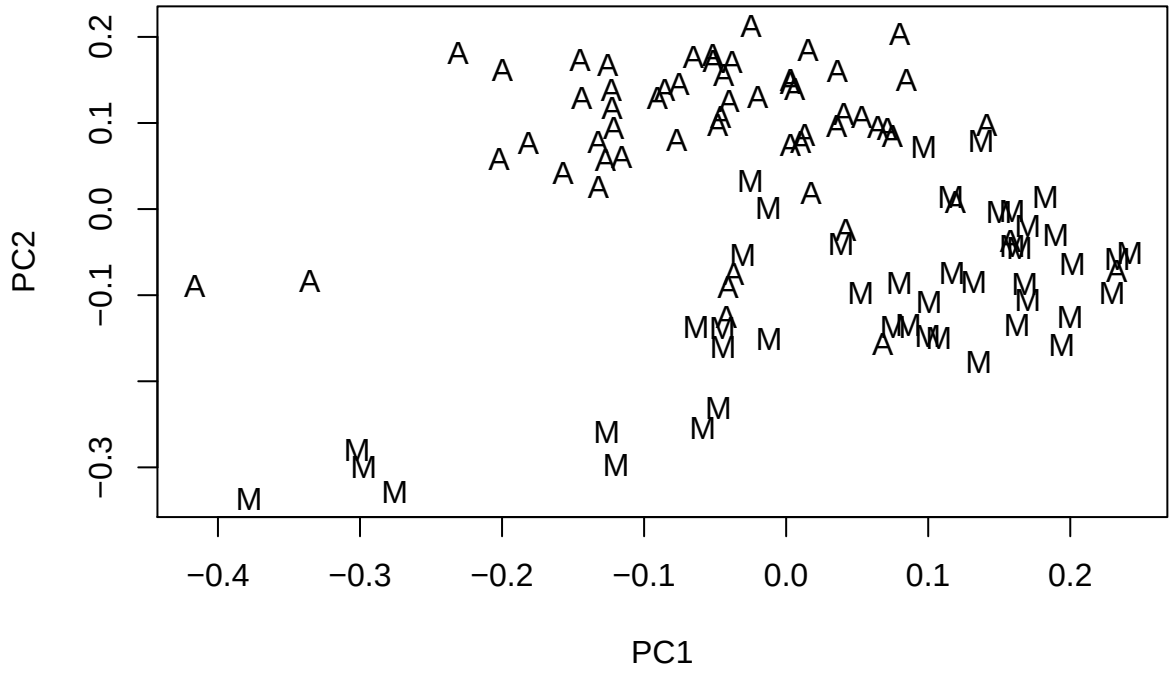
k-means clustering in the New York Times stories dataset.

```
set.seed(4268)
km.out <- kmeans(nyt.data[, -1], 2, nstart = 20)
km.out$cluster
```

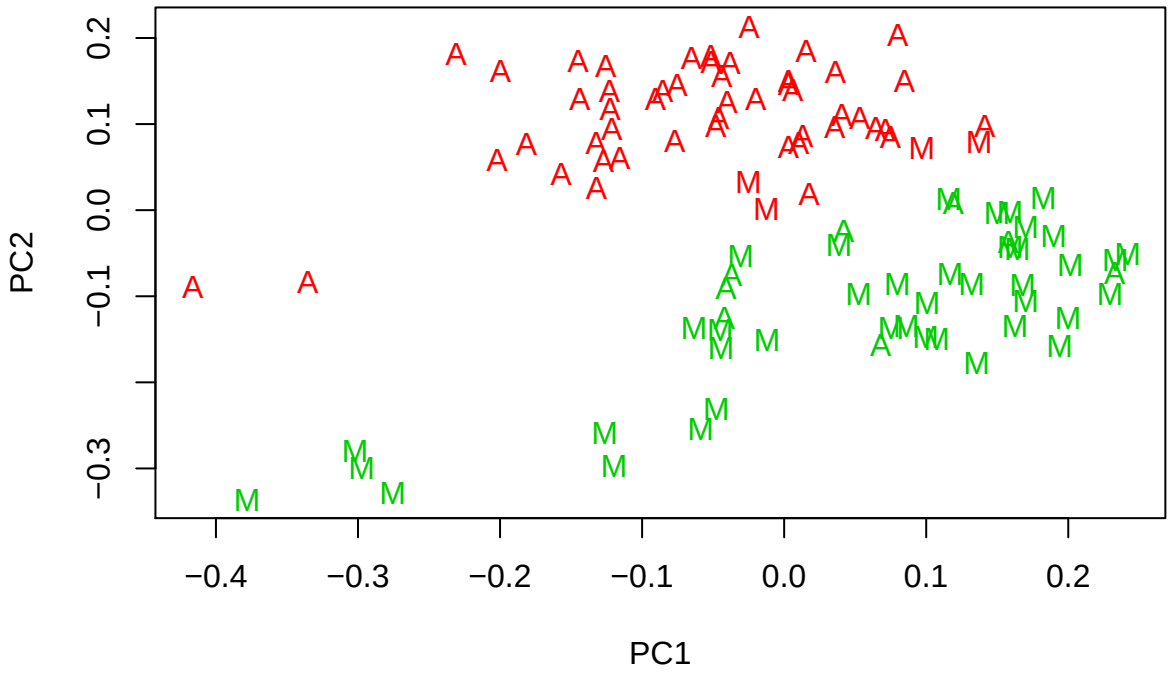
```
#>   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1
#>  [38] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1
#>  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2
```

```r
# Plot the clusters via Principal Components (PCA projections).

# PCA with true labels.
plot(pr.out$x[, 1:2], type = "n")
points(pr.out$x[nyt.data[, "class.labels"] == "art", 1:2], pch = "A")
points(pr.out$x[nyt.data[, "class.labels"] == "music", 1:2], pch = "M")
```




```r
# PCA with true labels but colored by k-means clustering.
plot(pr.out$x[, 1:2], type = "n")
points(pr.out$x[nyt.data[, "class.labels"] == "art", 1:2], pch = "A",
    col = (km.out$cluster + 1)[nyt.data[, "class.labels"] == "art"])
points(pr.out$x[nyt.data[, "class.labels"] == "music", 1:2], pch = "M",
    col = (km.out$cluster + 1)[nyt.data[, "class.labels"] == "music"])
```



Problem 4

Hierarcichal clustering in the New York Times stories dataset.

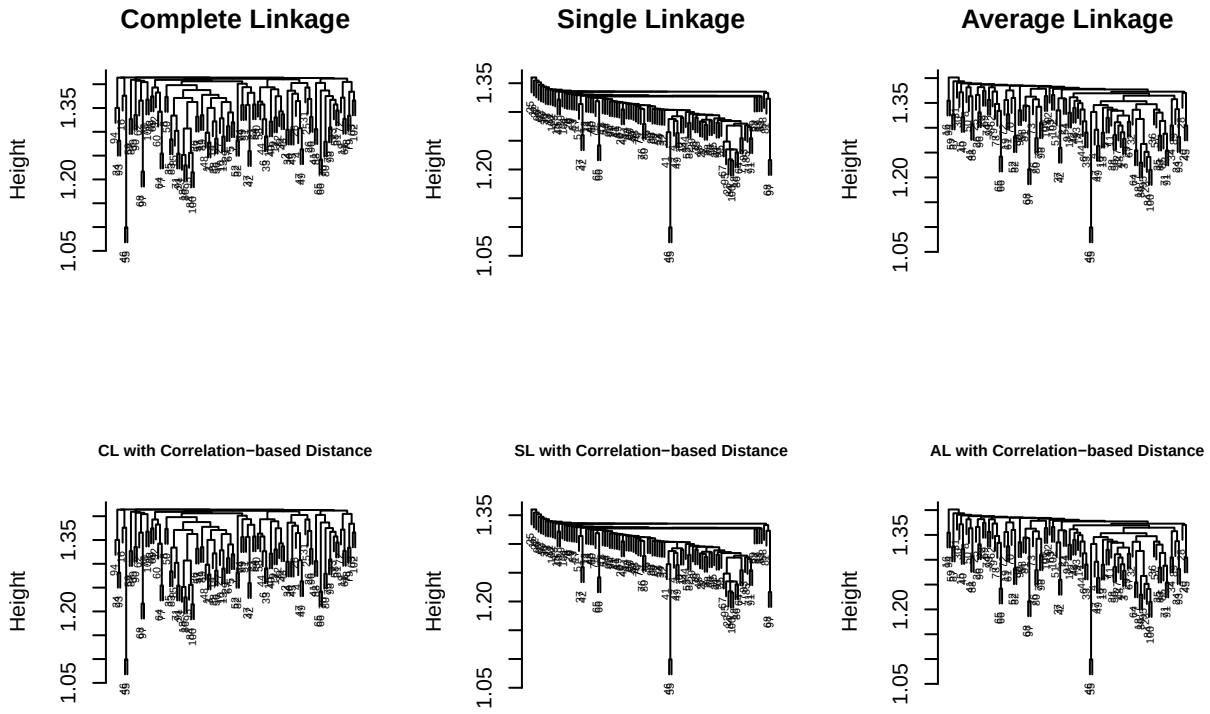
```
x <- nyt.data[, -1]

# Euclidean distances.
hc.complete <- hclust(dist(x), method = "complete")
hc.avg <- hclust(dist(x), method = "average")
hc.single <- hclust(dist(x), method = "single")

# Correlation-based distance.
dd <- as.dist(1 - cor(t(x)))
hc.cor.complete <- hclust(dist(x), method = "complete")
hc.cor.avg <- hclust(dist(x), method = "average")
hc.cor.single <- hclust(dist(x), method = "single")

par(mfrow = c(2, 3))
plot(hc.complete, main = "Complete Linkage", xlab = " ", sub = " ", cex = 0.5)
plot(hc.single, main = "Single Linkage", xlab = " ", sub = " ", cex = 0.5)
plot(hc.avg, main = "Average Linkage", xlab = " ", sub = " ", cex = 0.5)
plot(hc.cor.complete, cex.main = 0.7, main = "CL with Correlation-based Distance",
    xlab = " ", sub = " ", cex = 0.5)
plot(hc.cor.single, cex.main = 0.7, main = "SL with Correlation-based Distance",
    xlab = " ", sub = " ", cex = 0.5)
```

```
plot(hc.cor.avg, cex.main = 0.7, main = "AL with Correlation-based Distance",
    xlab = " ", sub = " ", cex = 0.5)
```




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

```
#>   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
#>  [38] 1 1 1 1 1 1 1 1 2 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
#>  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1
```

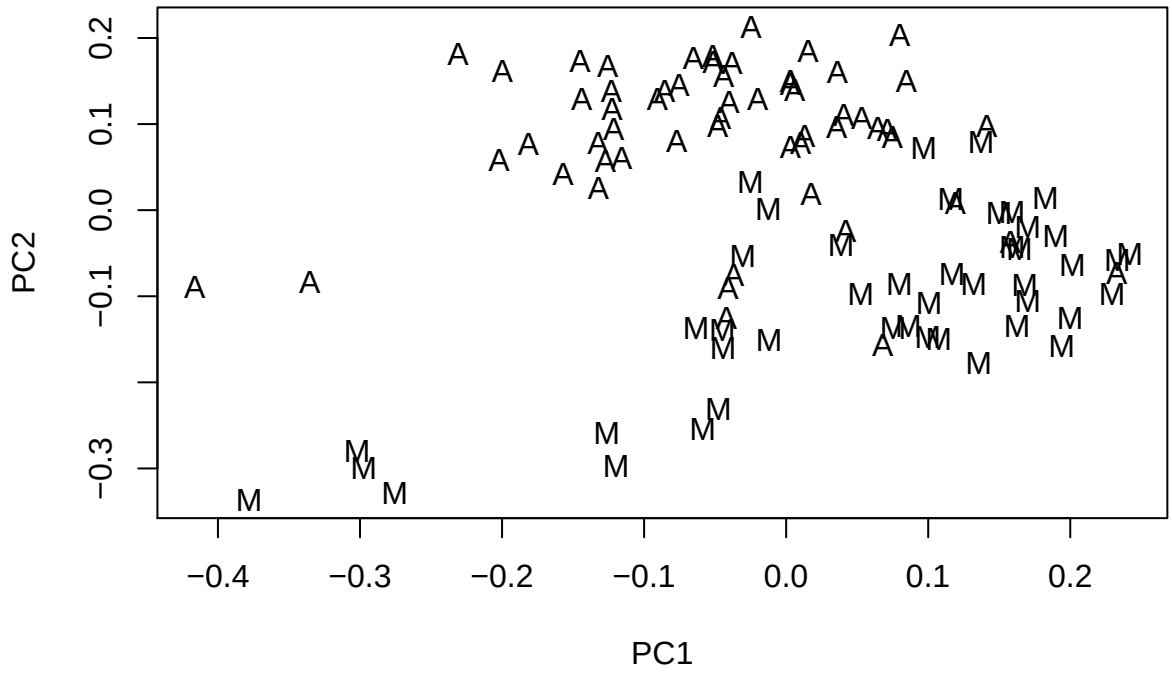
```
all.equal(cutree(hc.complete, 2), cutree(hc.single, 2), cutree(hc.avg,
    2), cutree(hc.cor.complete, 2), cutree(hc.cor.single, 2), cutree(hc.cor.avg,
    2))
```

```
#> [1] TRUE
```

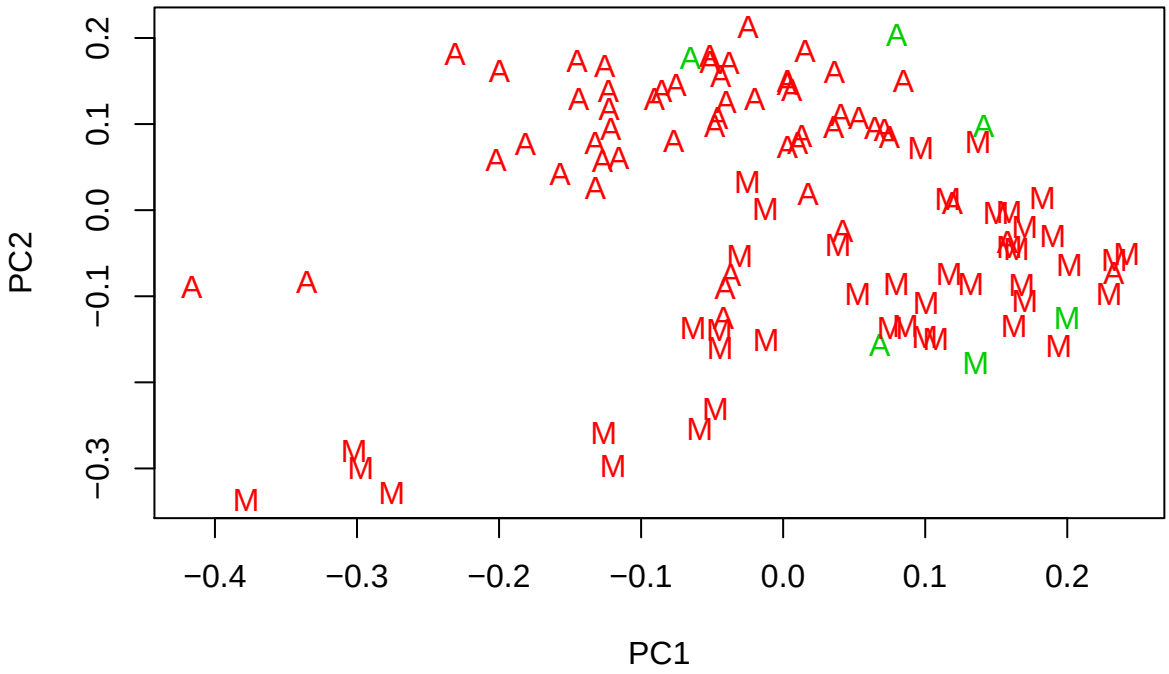
As is apparent, all the distances and linkage measures give the same split into two clusters.

```
hc.complete.clusters <- cutree(hc.complete, 2)
# Plot the clusters from Euclidean distance with complete linkage,
# via Principal Components (PCA projections).

# PCA with true labels.
plot(pr.out$x[, 1:2], type = "n")
points(pr.out$x[nyt.data[, "class.labels"] == "art", 1:2], pch = "A")
points(pr.out$x[nyt.data[, "class.labels"] == "music", 1:2], pch = "M")
```




```
# PCA with true labels but colored by k-means clustering.
plot(pr.out$x[, 1:2], type = "n")
points(pr.out$x[nyt.data[, "class.labels"] == "art", 1:2], pch = "A",
    col = (hc.complete.clusters + 1)[nyt.data[, "class.labels"] == "art"])
points(pr.out$x[nyt.data[, "class.labels"] == "music", 1:2], pch = "M",
    col = (hc.complete.clusters + 1)[nyt.data[, "class.labels"] == "music"])
```



It is clearly visible that the Hierarcichal clustering is a lot worse than k-means clustering when classifying into 2 clusters, on this data.