Lab 2: Clustering

Jonathan Bryan

August 31, 2018

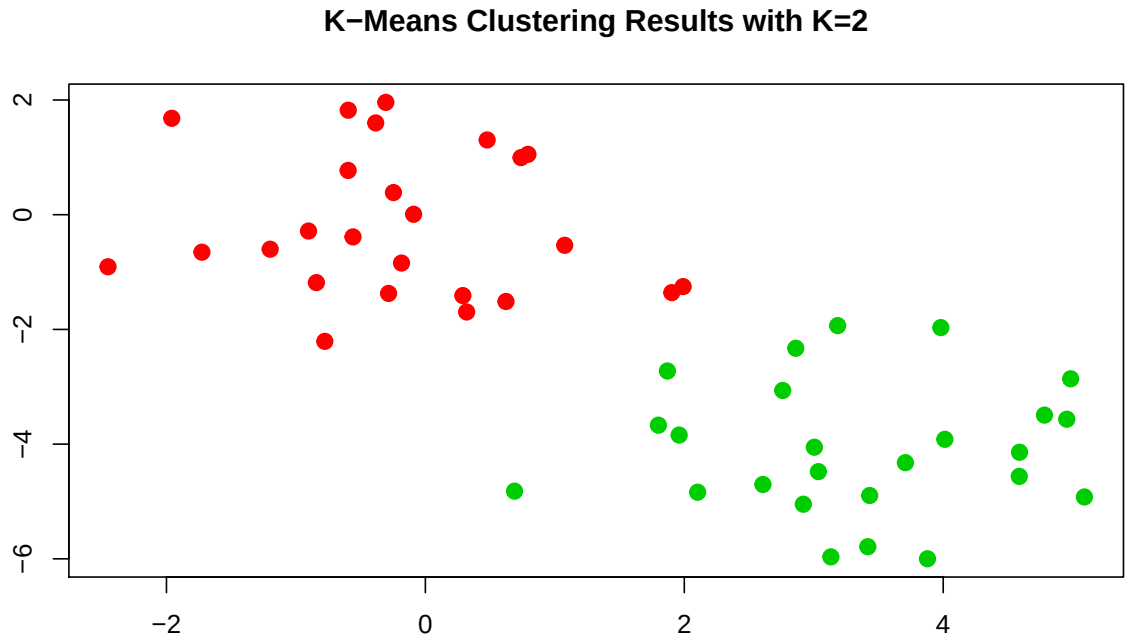
10.5.1 K-Means Clustering

```
#simulated data generation
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
```

```
#k-means clustering
km.out = kmeans(x,2,nstart = 20)
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 1 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
```

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



```r
#k=3 clustering
set.seed(4)
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
## [36] 2 2 2 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"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
#nstart controls number of k-means run
set.seed(3)
km.out = kmeans(x,3,nstart = 1)
km.out$tot.withinss
```

```
## [1] 104.3319
```

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

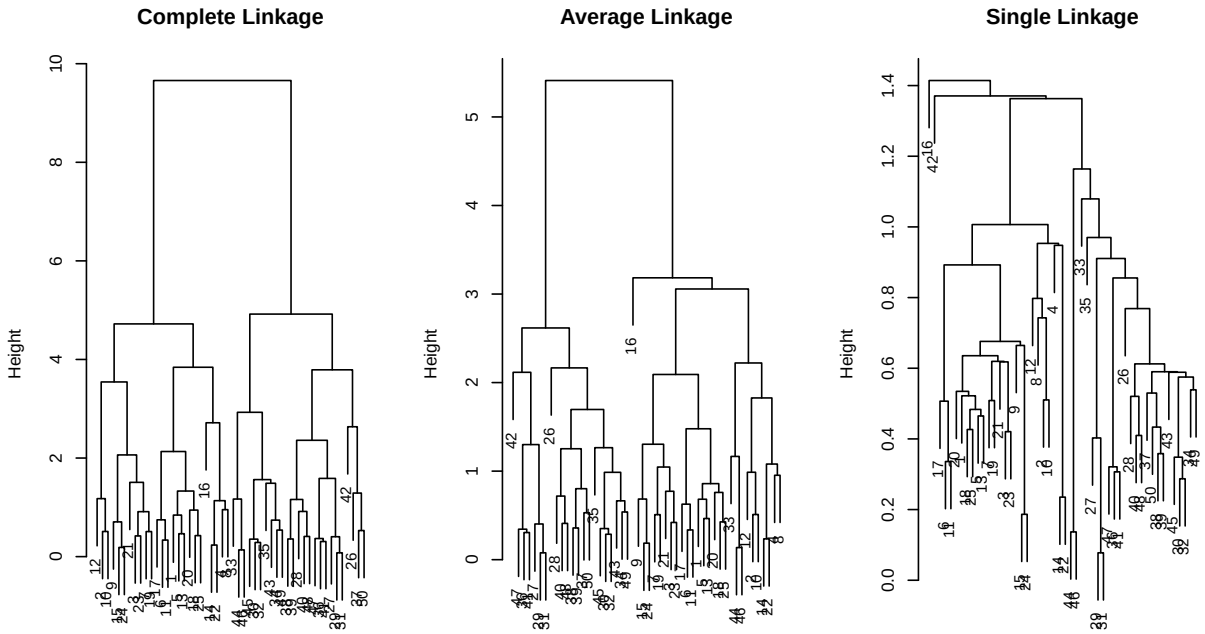
```
## [1] 97.97927
```

10.5.2 Hierarchical Clustering

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

```r
par(mfrow = c(1,3))
plot(hc.complete,
     main = "Complete Linkage",
     xlab = "",
     sub = "",
     cex = 0.9)
plot(hc.average,
     main = "Average Linkage",
     xlab = "",
     sub = "",
```

```
     cex = 0.9)
plot(hc.single,
     main = "Single Linkage",
     xlab = "",
     sub = "",
     cex = 0.9)
```




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

```
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 2 2 2 2 2 2 2 1 2 2
## [36] 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2
```

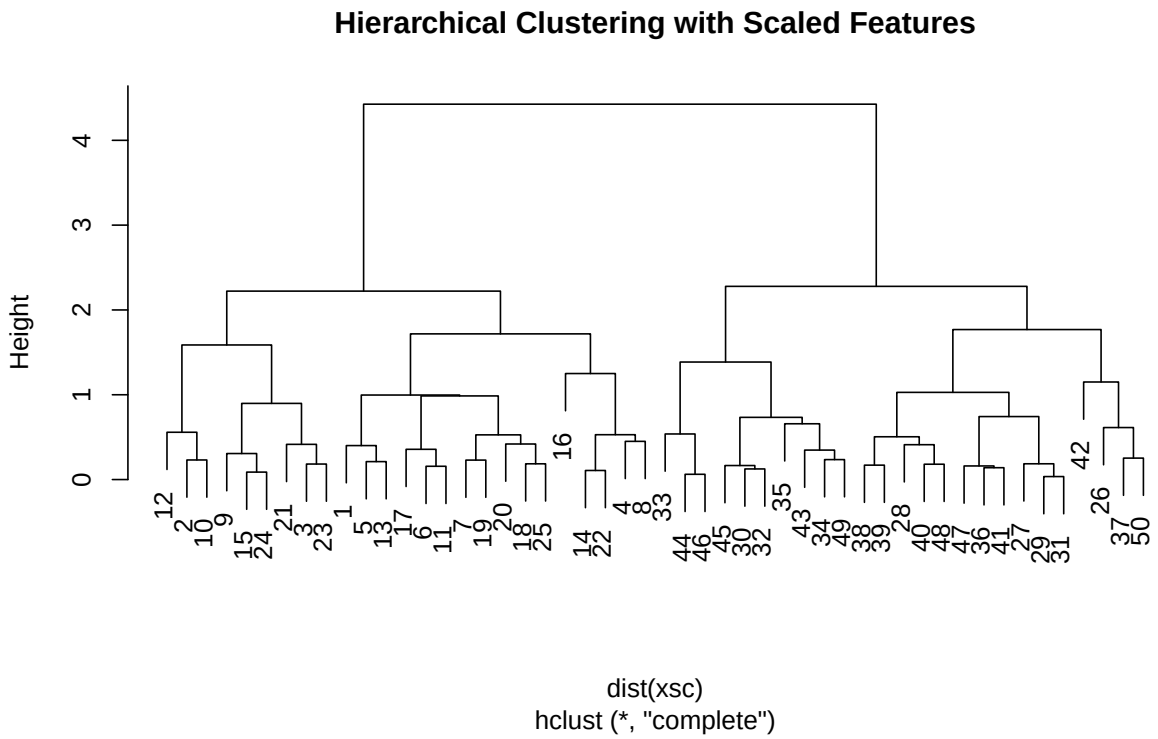
```
cutree(hc.single, 2)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 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 1
```

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

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3
## [36] 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3
```

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




```r
#correlation-based distance requires at least 3 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 =
```