R Notebook

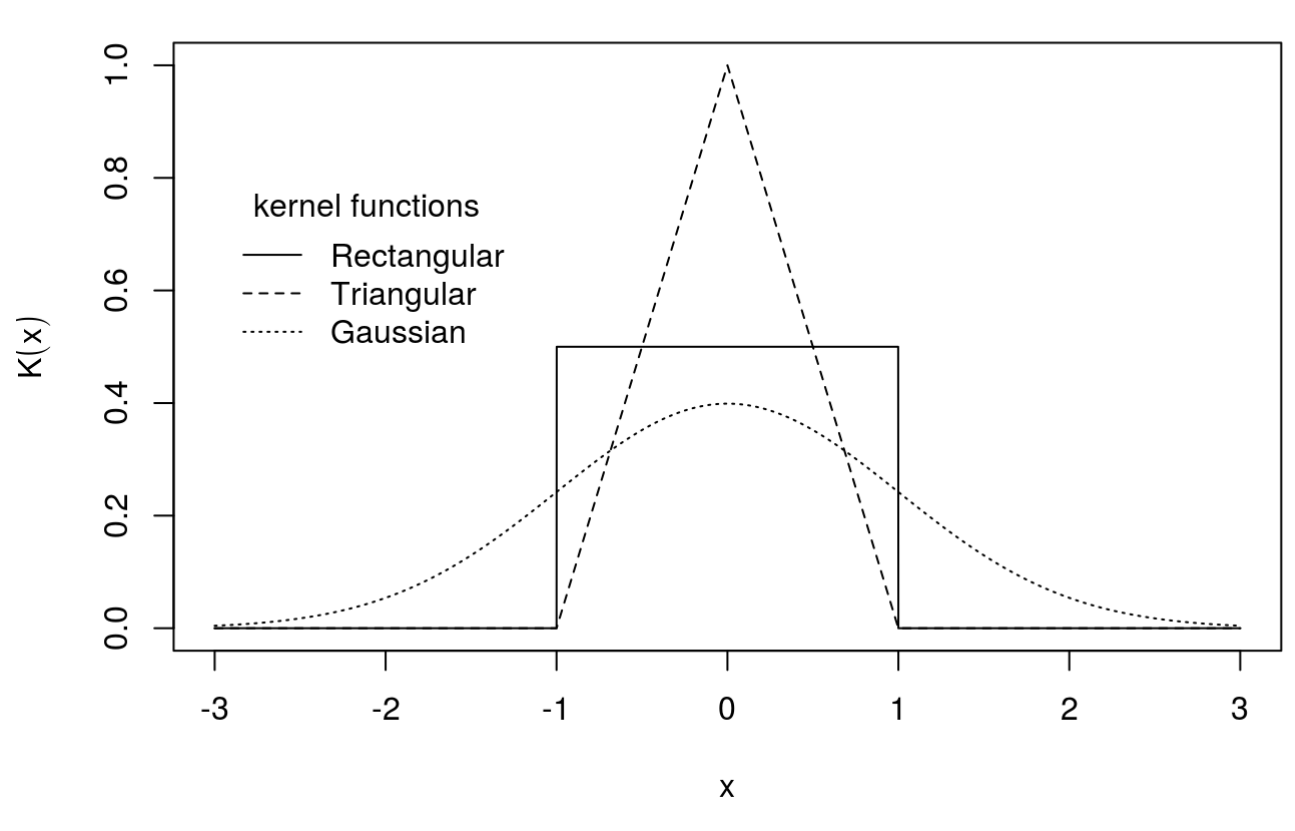
Octavio del Ser

10/11/2020

Question 1

Testing for different values of h. Variable declaration:

```
rec <- function(x) (abs(x) < 1) * 0.5
tri <- function(x) (abs(x) < 1) * (1 - abs(x))
gauss <- function(x) 1 / sqrt(2 * pi) * exp(-(x^2) / 2)
x <- seq(from = -3, to = 3, by = 0.001)
plot(x, rec(x), type = "l", ylim = c(0, 1), lty = 1,
     ylab = expression(K(x)))
lines(x, tri(x), lty = 2)
lines(x, gauss(x), lty = 3)
legend(-3, 0.8, legend = c("Rectangular", "Triangular", "Gaussian"),
       lty = 1:3, title = "kernel functions", bty = "n")
```

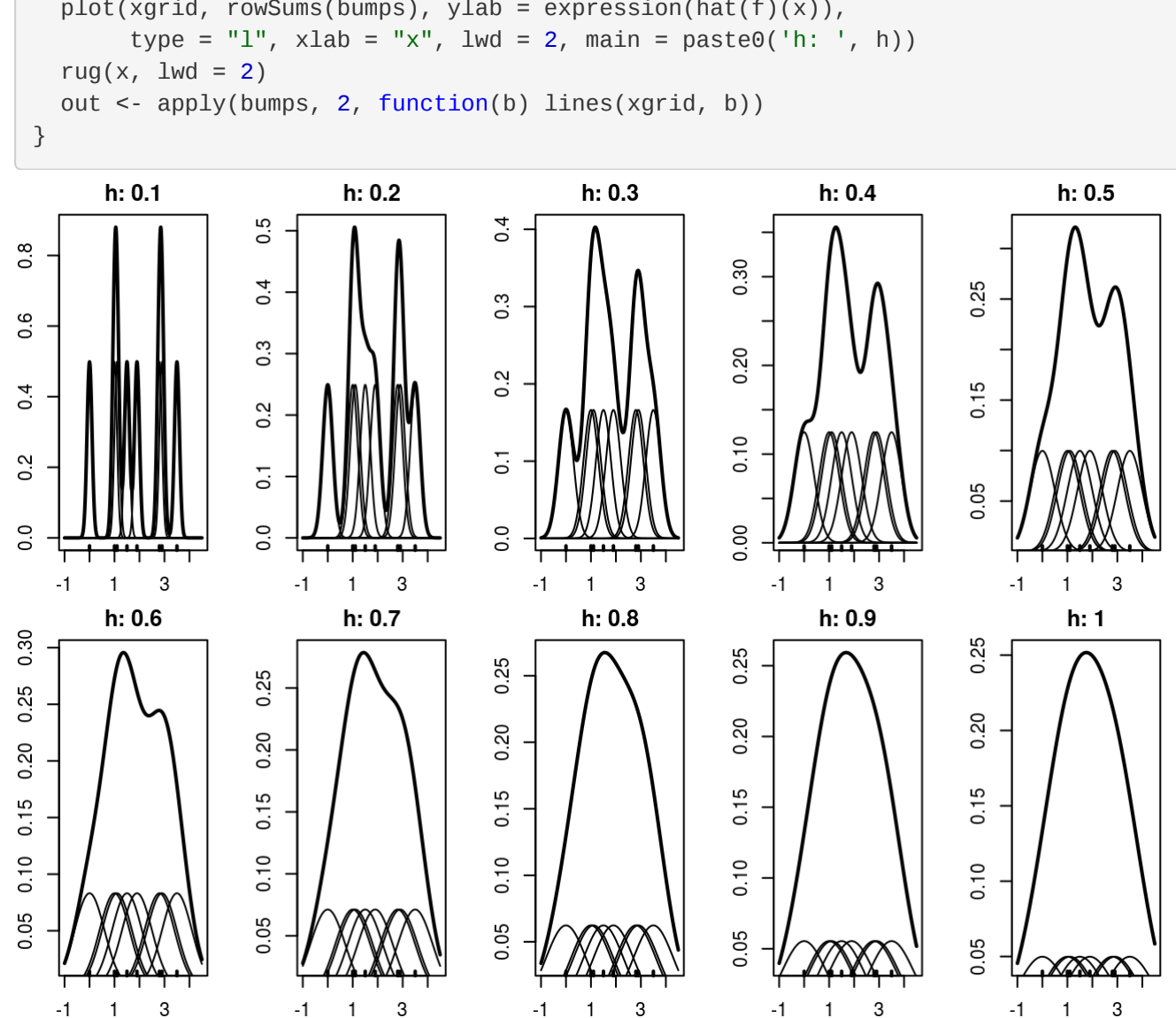



```
x <- c(0, 1, 1.1, 1.5, 1.9, 2.8, 2.9, 3.5)
n <- length(x)
xgrid <- seq(from = min(x) - 1, to = max(x) + 1, by = 0.01)
```

Testing different values: (h=0.1 -> 1 increments of .1

```
par(mfrow = c(2, 5))
par(mar = c(2, 2, 2, 2))

h_vals <- seq(0.1, 1, by = 0.1)
for (h in h_vals) {
  bumps <- sapply(x, function(a) gauss((xgrid - a) / h) / (n * h))
  plot(xgrid, rowSums(bumps), ylab = expression(hat(f)(x)),
       type = "l", xlab = "x", lwd = 2, main = paste0('h: ', h))
  rug(x, lwd = 2)
  out <- apply(bumps, 2, function(b) lines(xgrid, b))
}
```



As seen above the h value defines the trade-off between bias and variance, we want to choose a value of h as small as the data will allow, this way keeping bias low and variance as high as the prediction will allow. we can see as h increases the cure becomes smoother meaning variance is reduced but bias is large.

Question 2

a)

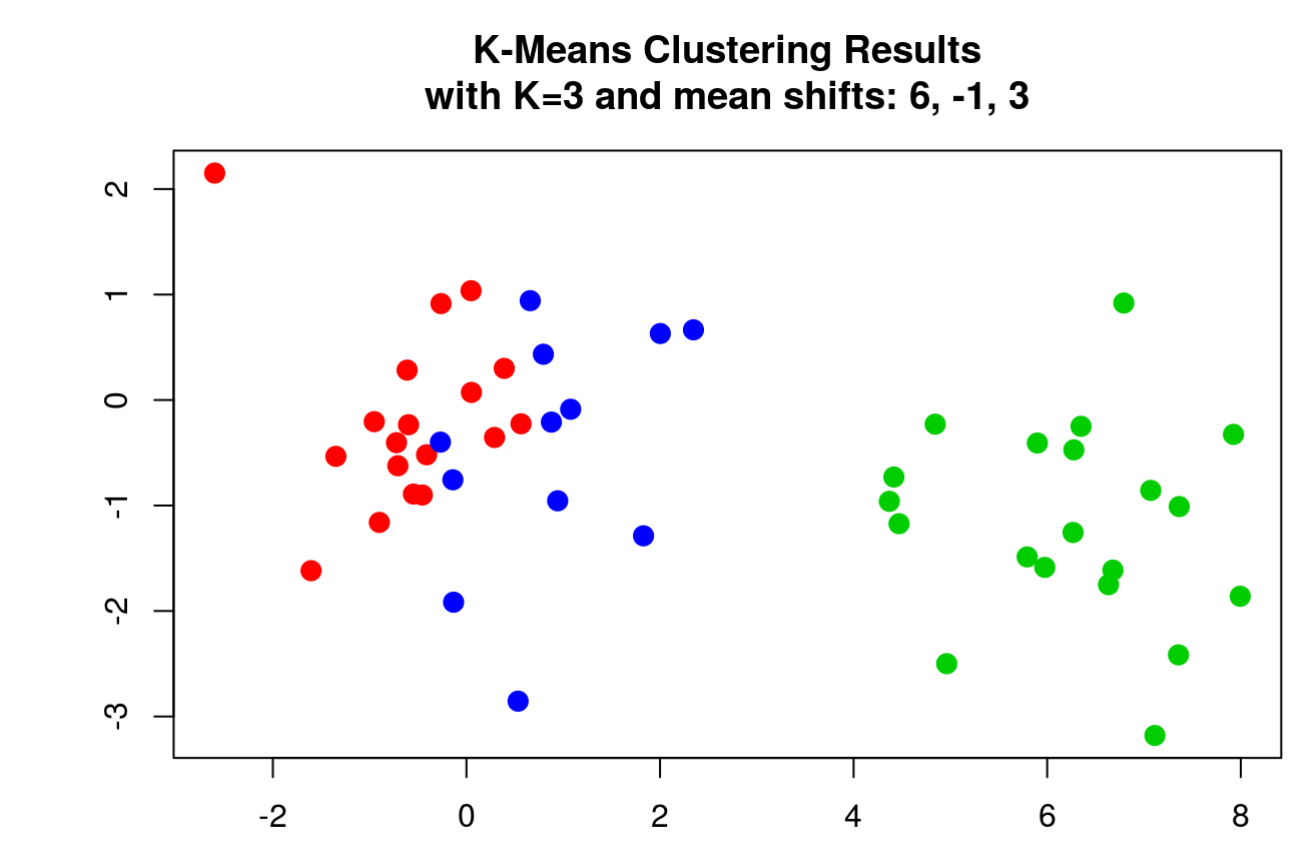
Generate three sets of 20 observation with their respective mean shift (60 total). 50 variables.

```
num_classes <- 3
variables <- 50
observations <- 20
set.seed(2394)
x <- matrix(rnorm(variables * num_classes), ncol = num_classes)
mean_shift <- sample(-10:10, num_classes, replace = FALSE)
for (n in 1:num_classes) {
  x[1:observations, n] <- x[1:observations, n] + mean_shift[n]
}
```

b)

Perform k-means

```
km.out <- kmeans(x, 3, nstart = 20)
plot(x, col = (km.out$cluster + 1), main = paste0("K-Means Clustering Results
with K=", num_classes, ' and mean shifts: ', paste(mean_shift, collapse = ', ')), xlab = "", ylab = "", pch = 20,
cex = 2)
```




```
km.out
```

```
K-means clustering with 3 clusters of sizes 18, 20, 12

Cluster means:
        [,1]       [,2]       [,3]
1 -0.5769918 -0.1618841  0.6451578
2  6.2271734 -1.1572885  3.0521206
3  0.8766245 -0.4829313 -1.0955948

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

Within cluster sum of squares by cluster:
[1] 35.42230 48.60477 32.58231
 (between_SS / total_SS =  84.3 %)

Available components:

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

```
km.out$tot.withinss
```

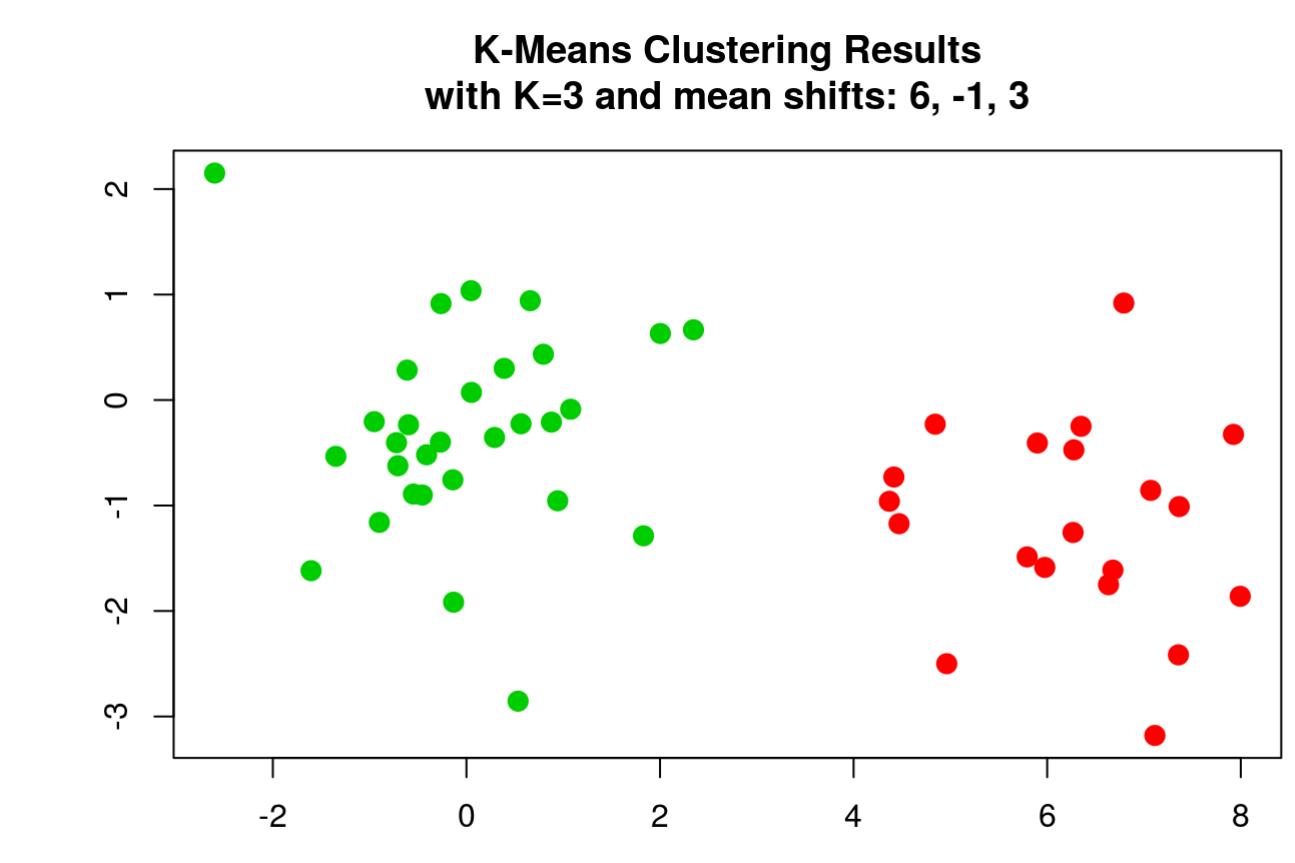
```
[1] 116.6094
```

We can observer that the classification was correct, k means identified the mean shift in the dataset. within cluster sum of squares by cluster result of 84.3%

c)

k-means -> k=2

```
km.out <- kmeans(x, 2, nstart = 20)
plot(x, col = (km.out$cluster + 1), main = paste0("K-Means Clustering Results
with K=", num_classes, ' and mean shifts: ', paste(mean_shift, collapse = ', ')), xlab = "", ylab = "", pch = 20,
cex = 2)
```




```
km.out
```

```
K-means clustering with 2 clusters of sizes 20, 30

Cluster means:
         [,1]      [,2]        [,3]
1 6.227173396 -1.157288  3.05212059
2 0.004454728 -0.290303 -0.05114327

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

Within cluster sum of squares by cluster:
[1]  48.60477 105.77791
 (between_SS / total_SS =  79.2 %)

Available components:

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

```
km.out$tot.withinss
```

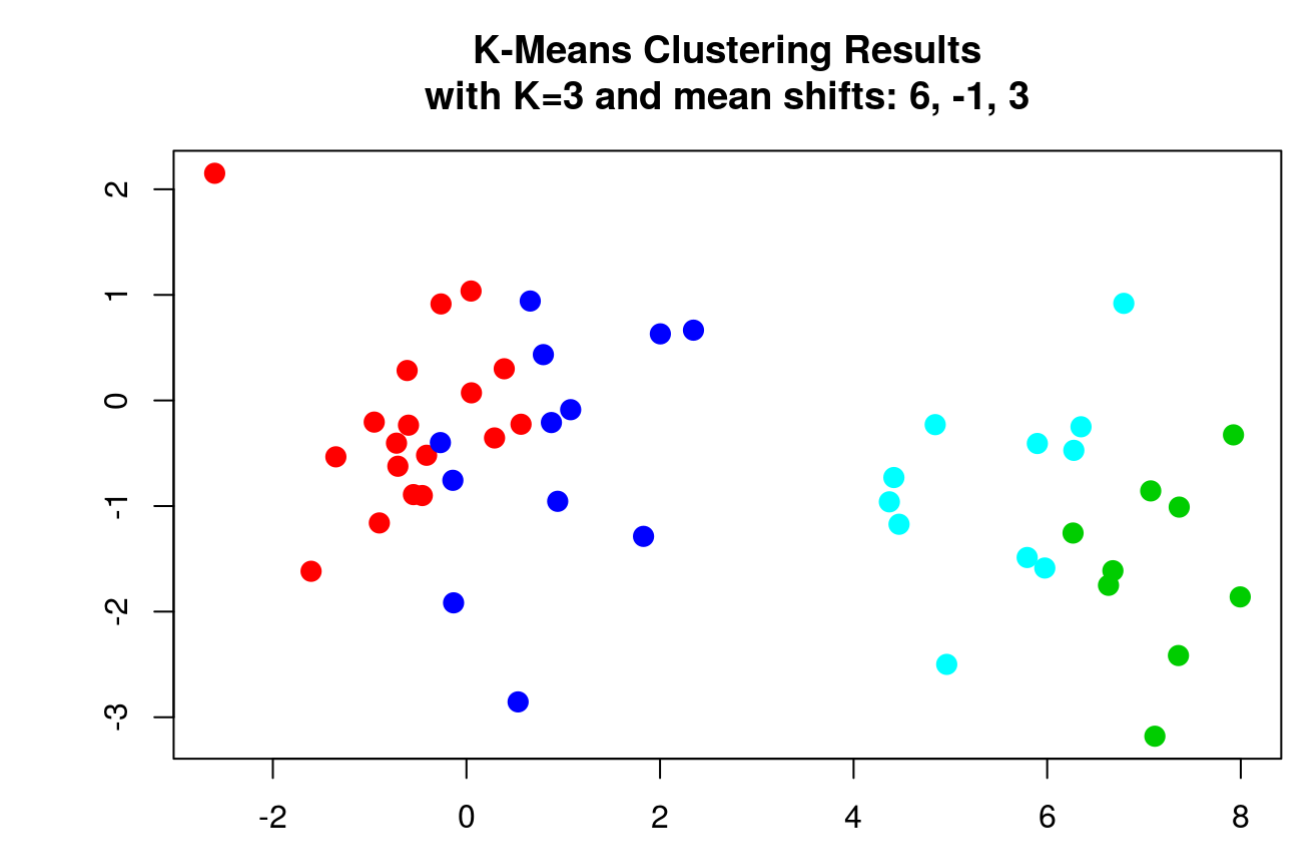
```
[1] 154.3827
```

(between_SS / total_SS = 79.2 %) higher bias, lower variance.

d)

k-means k=4

```
km.out <- kmeans(x, 4, nstart = 20)
plot(x, col = (km.out$cluster + 1), main = paste0("K-Means Clustering Results
with K=", num_classes, ' and mean shifts: ', paste(mean_shift, collapse = ', ')), xlab = "", ylab = "", pch = 20,
cex = 2)
```




```
km.out
```

```
K-means clustering with 4 clusters of sizes 18, 9, 12, 11

Cluster means:
        [,1]       [,2]       [,3]
1 -0.5769918 -0.1618841  0.6451578
2  7.1555500 -1.5854634  3.3145909
3  0.8766245 -0.4829313 -1.0955948
4  5.4675925 -0.8069635  2.8373721

Clustering vector:
 [1] 4 2 2 2 2 2 2 4 4 4 2 2 4 4 4 4 2 4 4 4 3 3 3 1 1 1 1 1 3 1 1 1 1 1 3 3 3 1
[39] 3 3 1 1 3 1 3 1 1 3 1 1

Within cluster sum of squares by cluster:
[1] 35.42230 11.46924 32.58231 18.90467
 (between_SS / total_SS =  86.8 %)

Available components:

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

```
km.out$tot.withinss
```

```
[1] 98.37853
```

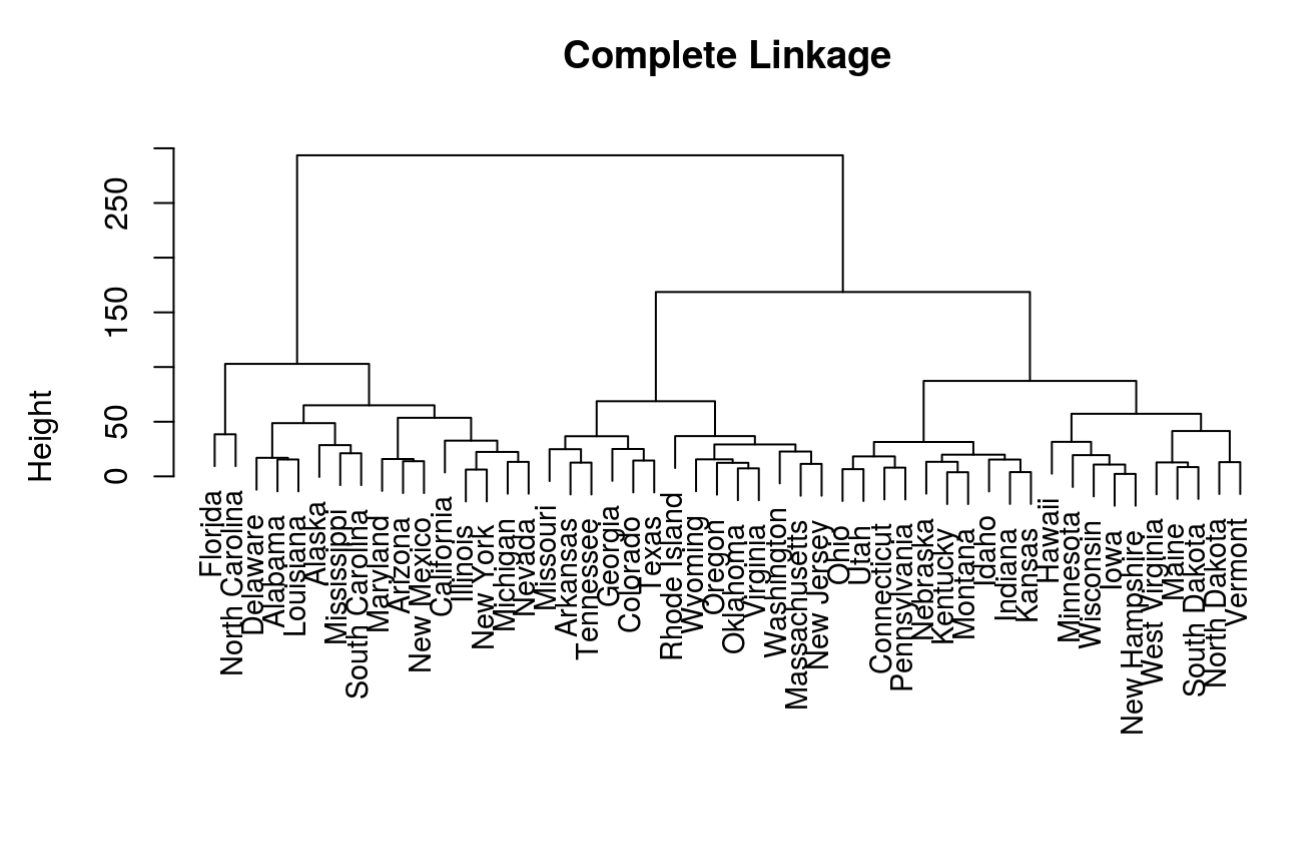
(between_SS / total_SS = 86.8 %) lower bias than k=2 but higher variance

Question 3

a)

Load Dataset and perform complete linkage HC:

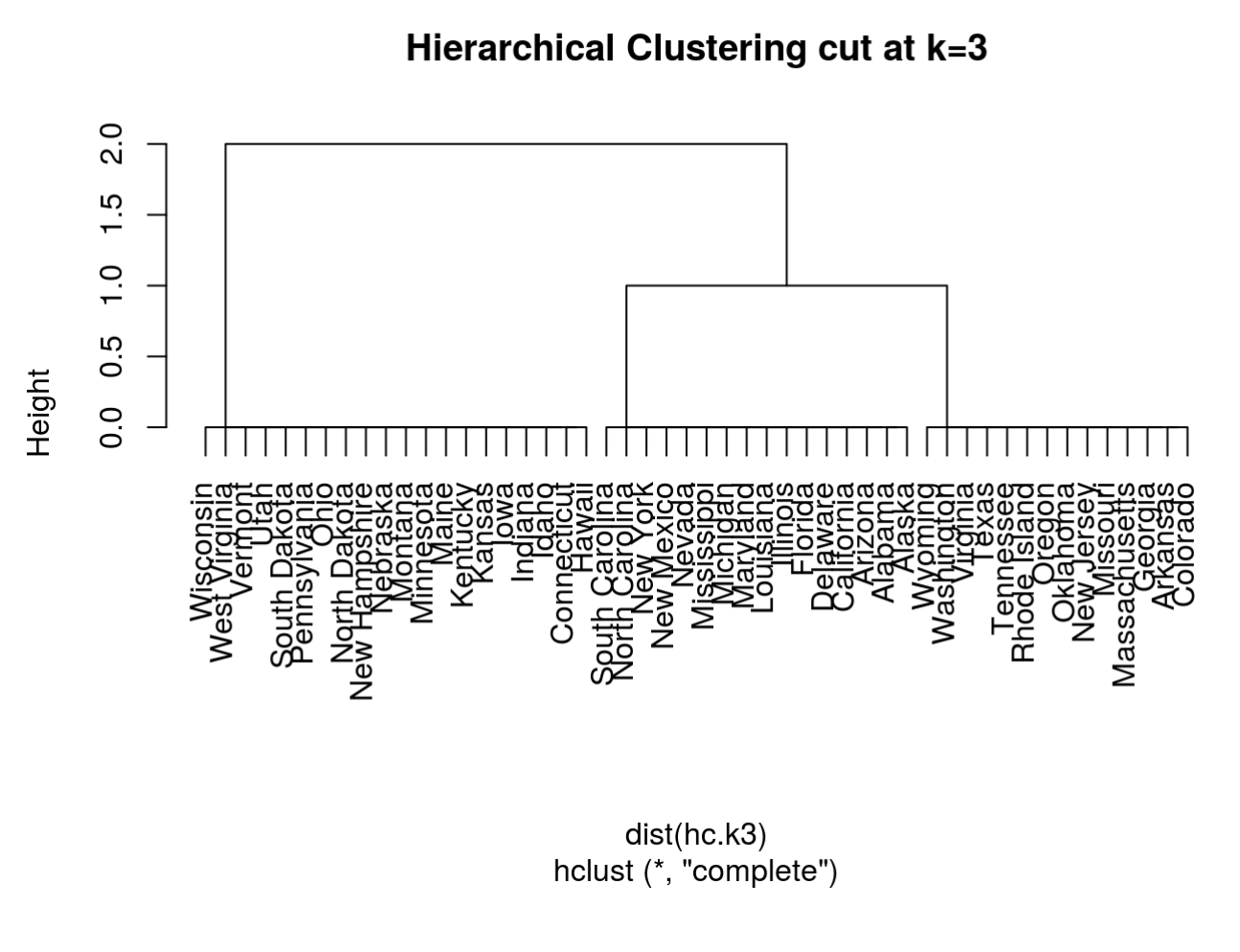
```
x <- USArrests
hc.complete <- hclust(dist(x), method = "complete")
plot(hc.complete, main = "Complete Linkage", xlab = "", sub = "", cex = .9)
```



b)

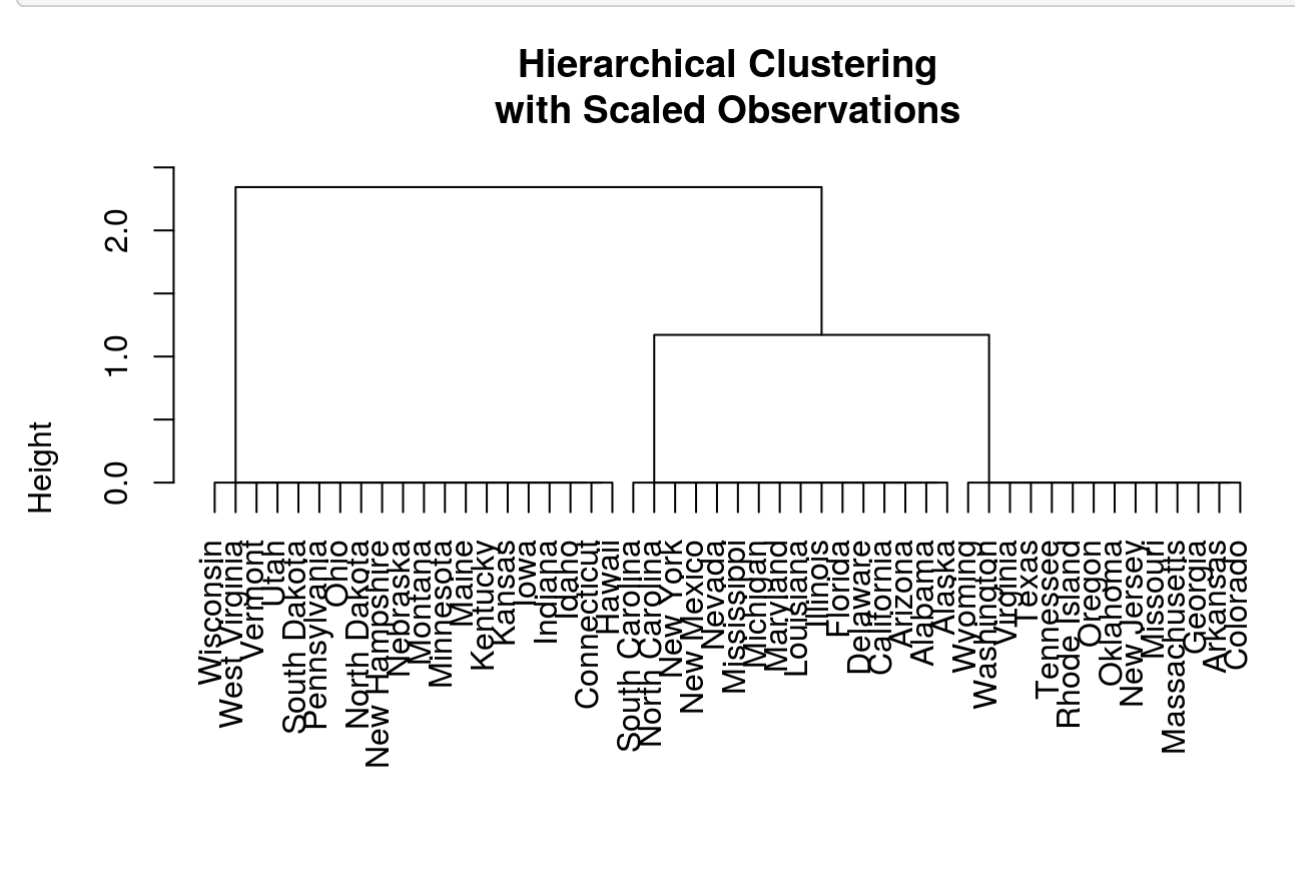
Cut dendrogram at clusters=3

```
hc.k3 <- cutree(hc.complete, 3)
plot(hclust(dist(hc.k3), method = "complete"), main = "Hierarchical Clustering cut at k=3")
```



c)

```
xsc <- scale(hc.k3)
plot(hclust(dist(xsc), method = "complete"), main = "Hierarchical Clustering
with Scaled Observations")
```



d)

Scaling the dataset makes the variances equal as more weight is not placed on variables with greater values. Variables should be scaled before inter-observation dissimilarities to avoid placing more weight on the variance in variable values. Scaling data also rarely hurts hence it is a good practice in general.