Lab 16

quan le

2022-10-18

K-Means Clustering

```
library(ggplot2)
library(ISLR)
library(kknn)
```

1. Data set 1 - Simulated Data

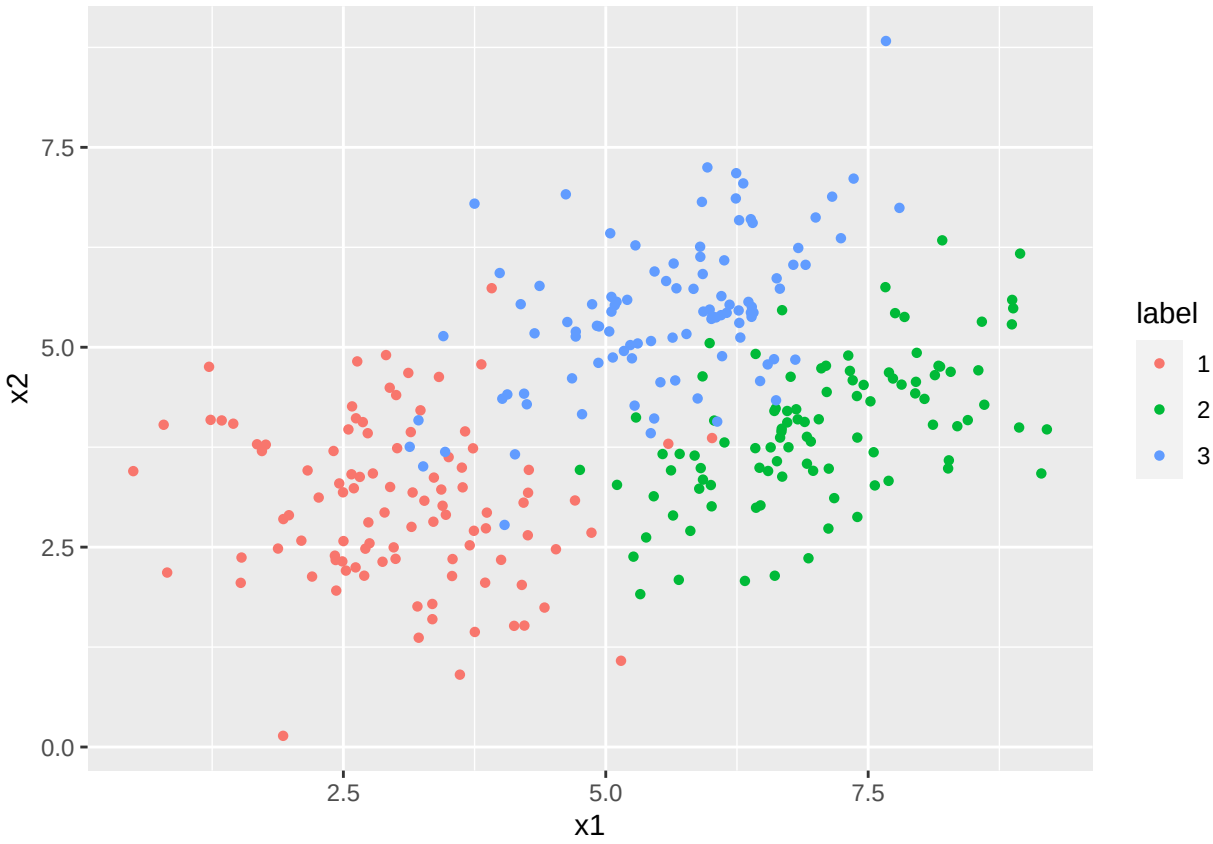
- small simulated data set to demonstrate concepts with k-means clustering

Simulate data: generate data from a mixture of three normal distribution

```
n = 300
mu1 = c(3,3)
mu2 = c(7,4)
mu3 = c(5.5,5.5)
Sig = matrix(c(1,.5,.5,1),2,2)
x1 = t(matrix(mu1,2,n/3)) + matrix(rnorm(n*2/3),n/3,2)
xx = matrix(rnorm(n*2/3),n/3,2)
x2 = t(matrix(mu2,2,n/3)) + xx%*%chol(Sig)
xx = matrix(rnorm(n*2/3),n/3,2)
x3 = t(matrix(mu3,2,n/3)) + xx%*%chol(Sig)
X = rbind(x1,x2,x3)
Y = c(rep(1,n/3),rep(2,n/3),rep(3,n/3))
Data = cbind(X,Y)
Data = data.frame(Data)
colnames(Data) = c("x1","x2","label")
Data$label = factor(Data$label)
```

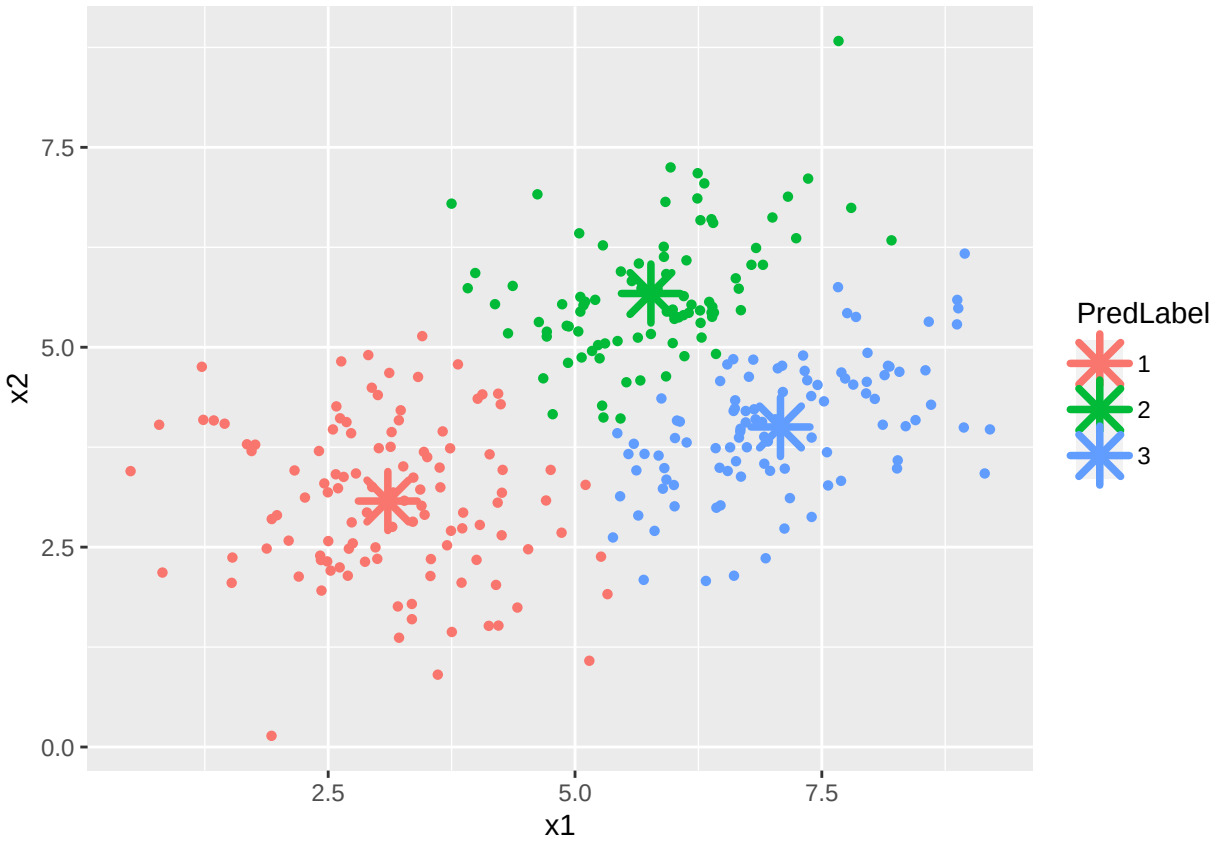
Plot with true labels

```
ggplot(data = Data) +
  geom_point(mapping = aes(x = x1,y = x2,color = label),pch = 16)
```



Apply k-means

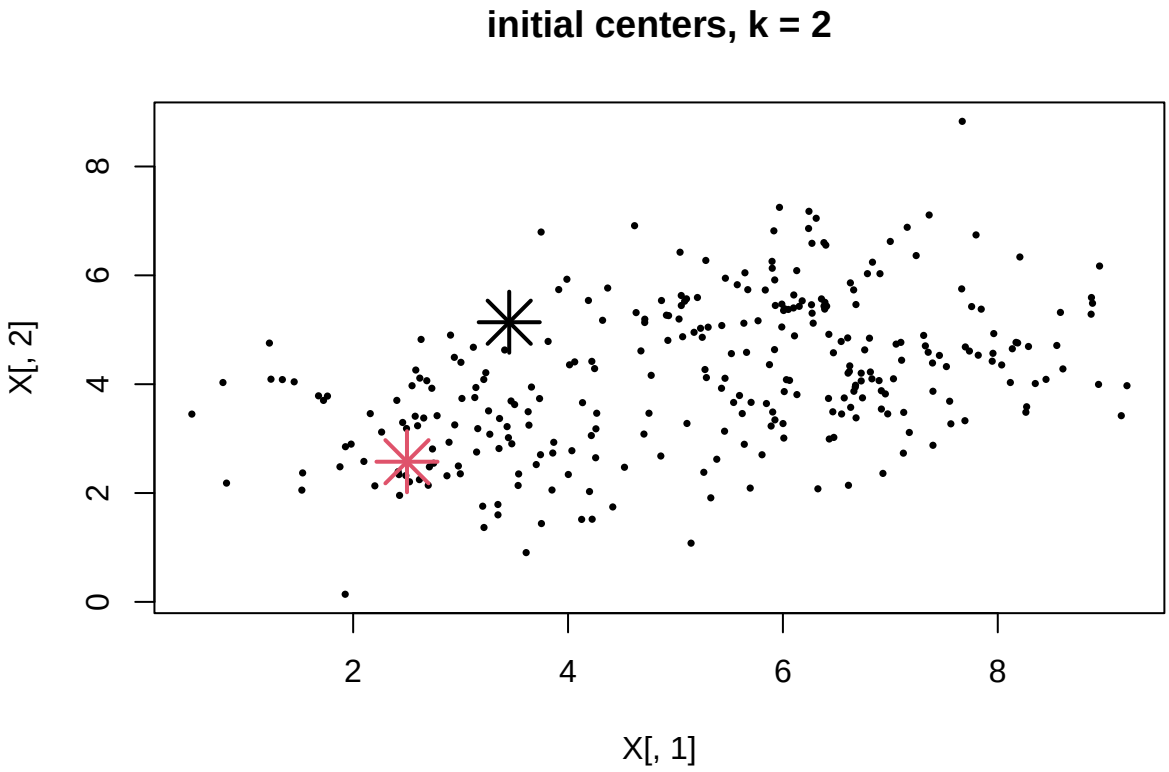
```
k = 3
km = kmeans(X,centers=k)
gd = data.frame(km$centers)
gd$label = rownames(gd)
colnames(gd) = c("x1","x2","label")
Data$PredLabel = factor(km$cluster)
ggplot() +
    geom_point(data = Data,mapping = aes(x = x1,y = x2,color = PredLabel), pch = 16)  +
    geom_point(gd,mapping = aes(x = x1,y = x2,color= factor(label)),size = 6, shape = 8, stroke = 2)
```

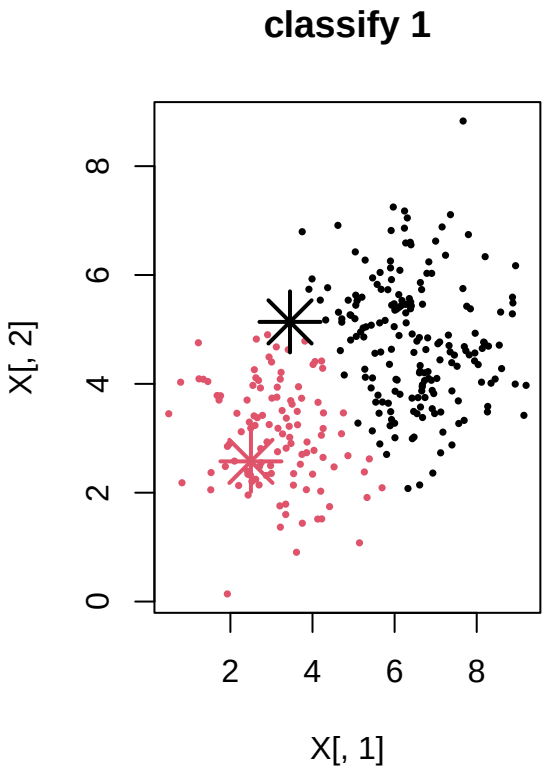


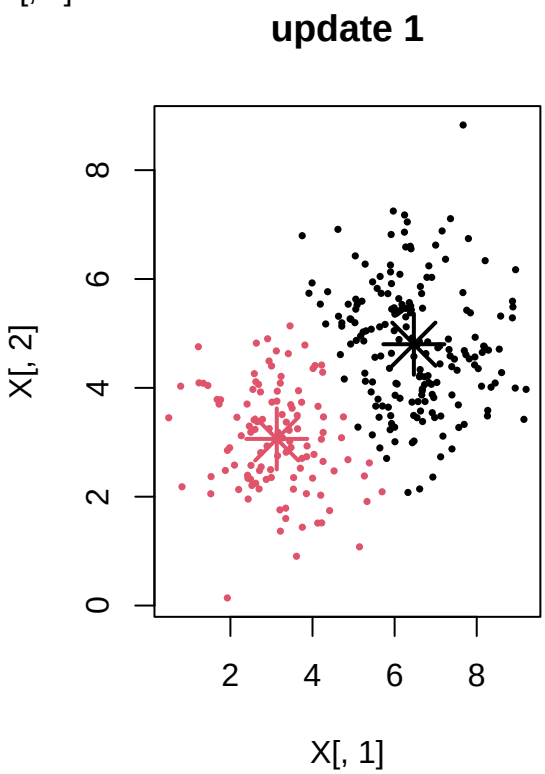
Varying k

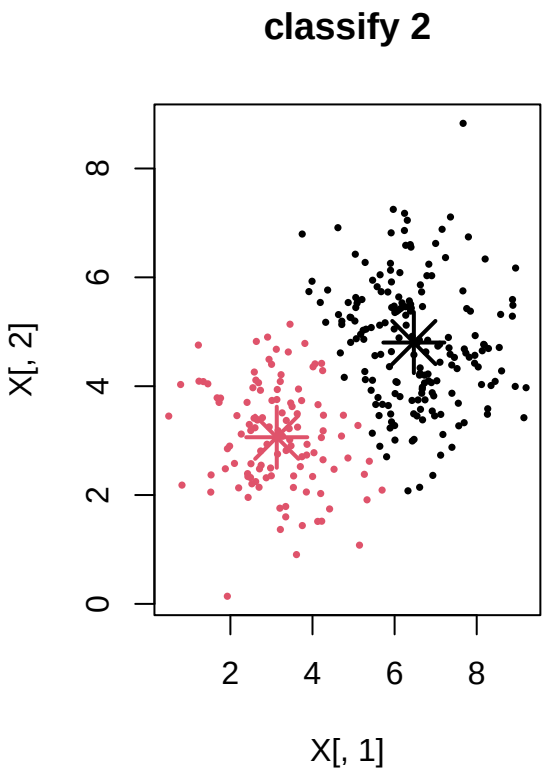
```
for (k in 2:5) {
  n = nrow(X)
  cens = X[sample(1:n,k),]
  par(mfrow=c(1,1))
  plot(X[,1],X[,2],pch=16, cex = 0.5, main=paste("initial centers, k =", k))
  points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd = 2)
  par(mfrow=c(1,2))
  for(i in 1:5) {
    oldcen = cens
    km = kmeans(X,centers=cens,iter.max=1,nstart=1,algorithm="MacQueen")
    plot(X[,1],X[,2],col=km$cluster,pch=16, cex = 0.5, main=paste("classify", i))
    points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd= 2)
    cens = km$centers
    plot(X[,1],X[,2],col=km$cluster,pch=16,cex=0.5, main=paste("update", i))
    points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd= 2)
    ind = sum(diag((oldcen-cens)%*%t(oldcen-cens)))
  }
}
```

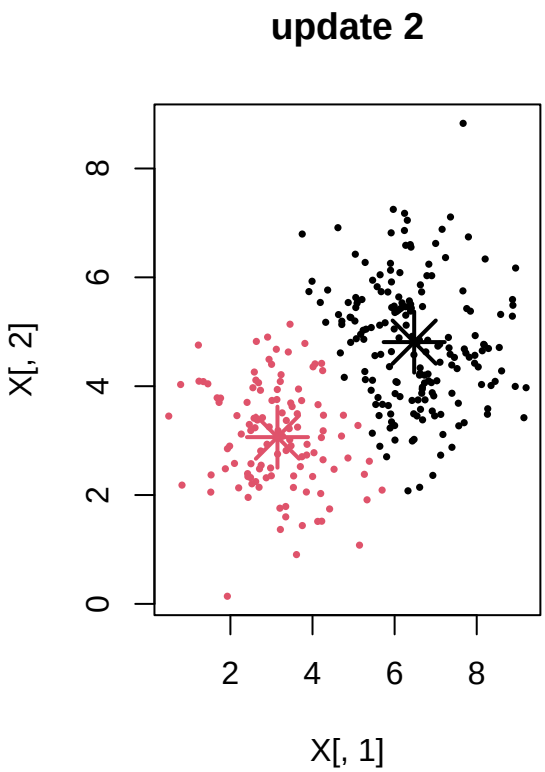
```
## Warning: did not converge in 1 iteration
```

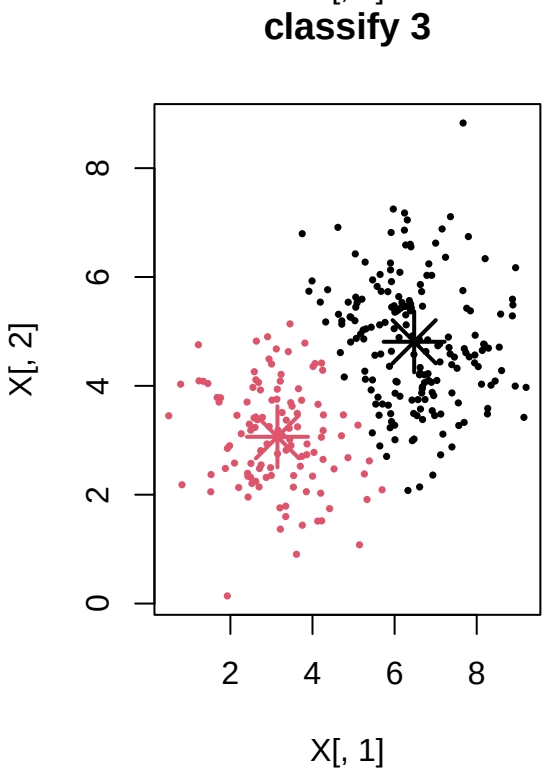

initial centers, k = 2
X[, 2]
X[, 1]
0
2
4
6
8

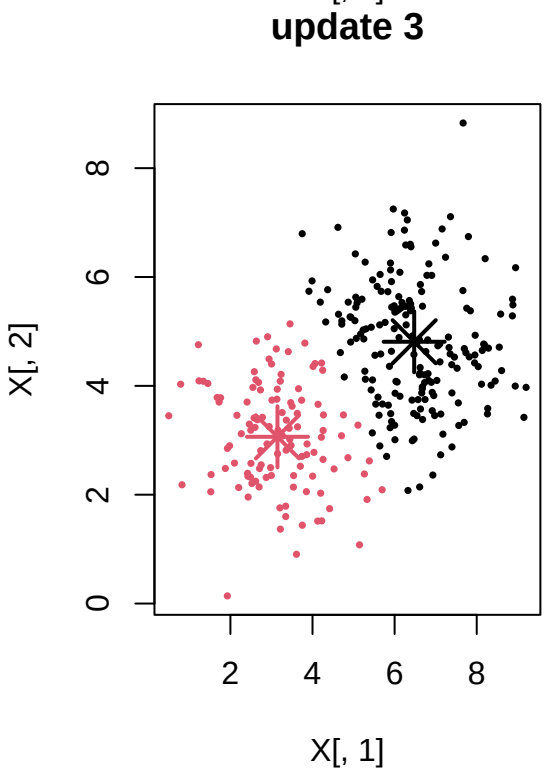

classify 1
X[, 2]
X[, 1]
0
2
4
6
8

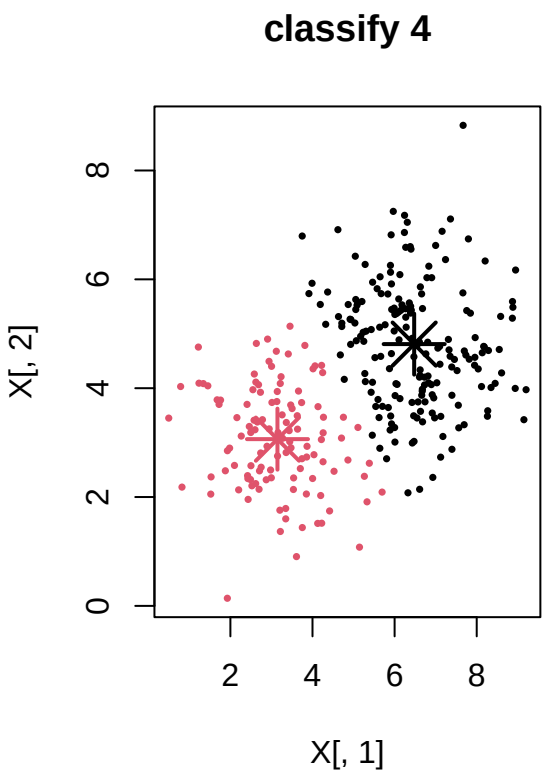

update 1
X[, 2]
X[, 1]
0
2
4
6
8

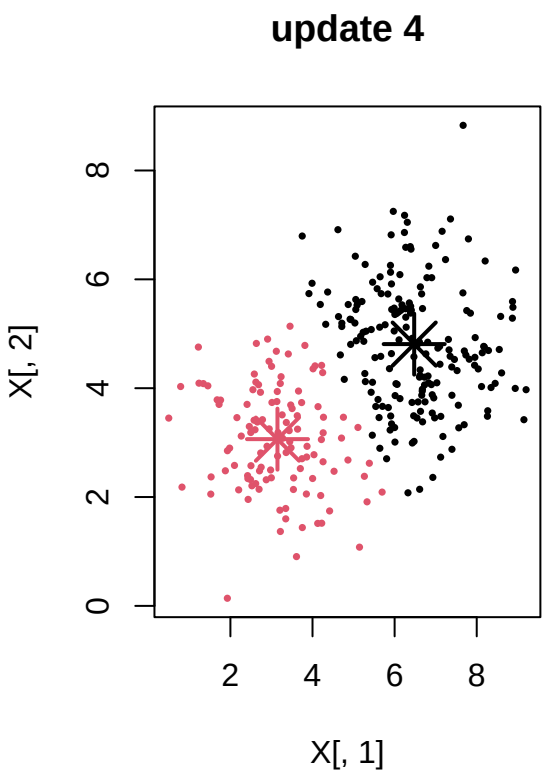

classify 2
X[, 2]
X[, 1]
0
2
4
6
8

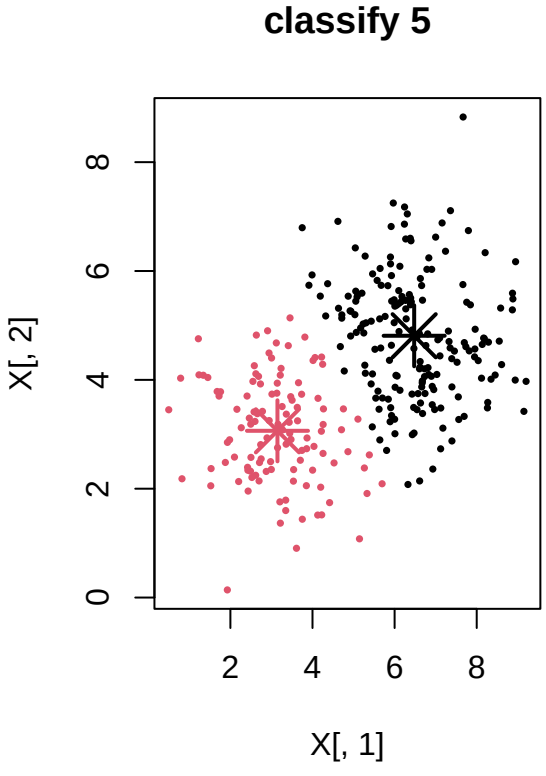

update 2
X[, 2]
X[, 1]
0
2
4
6
8

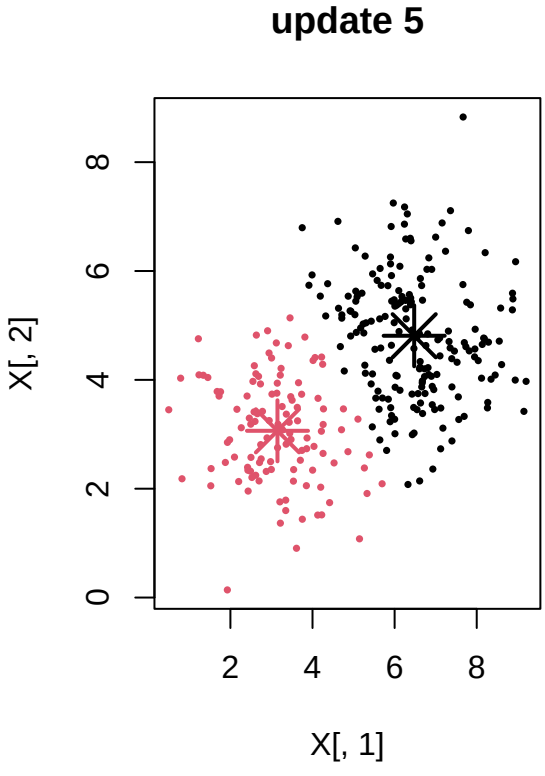

classify 3
X[, 2]
X[, 1]
0
2
4
6
8


update 3
X[, 2]
X[, 1]
0
2
4
6
8

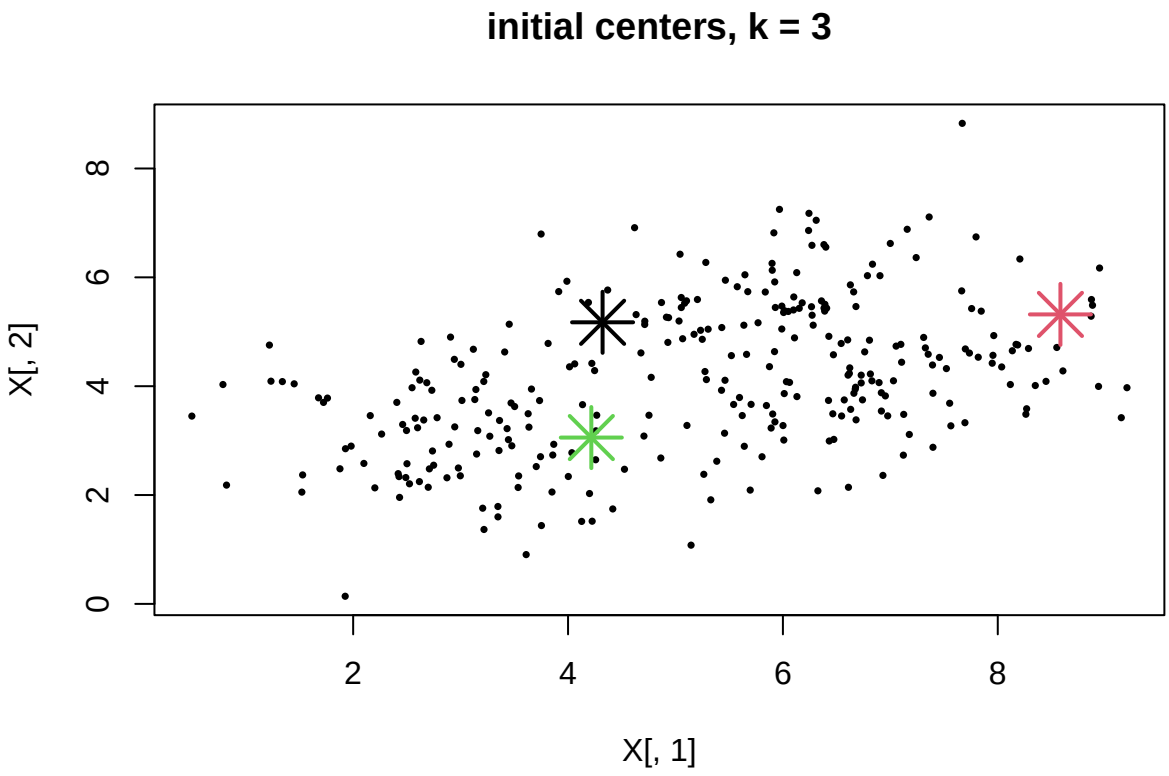




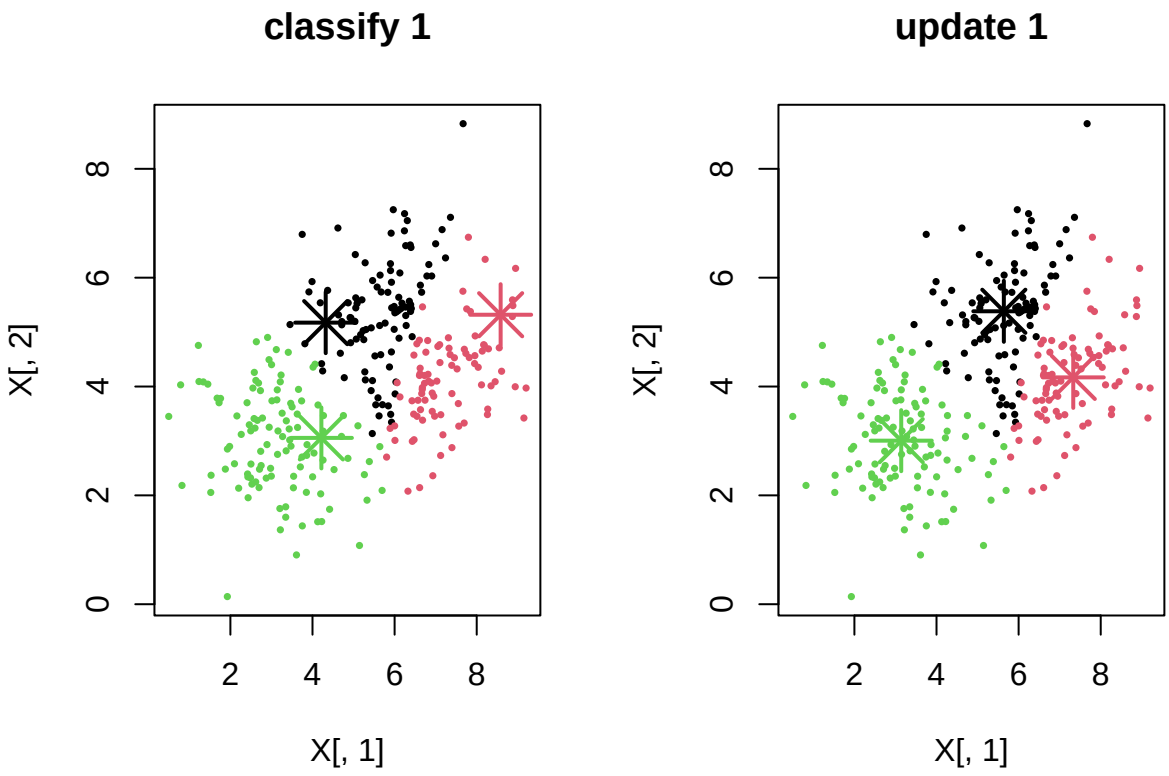


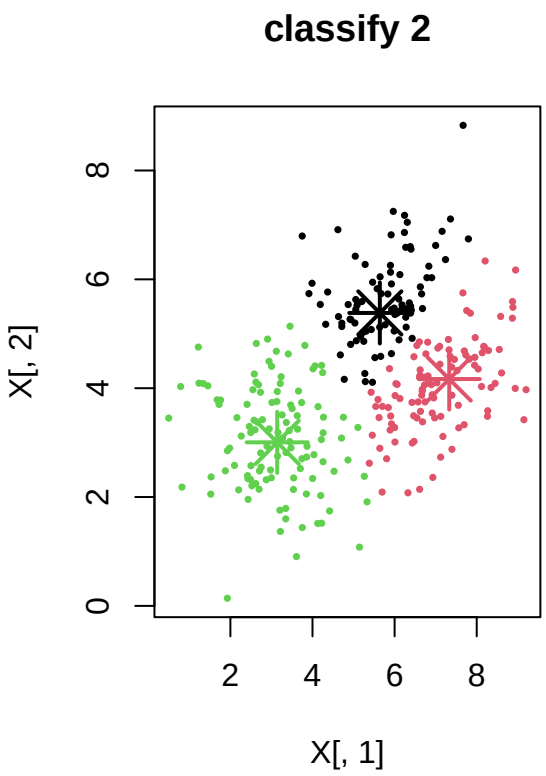


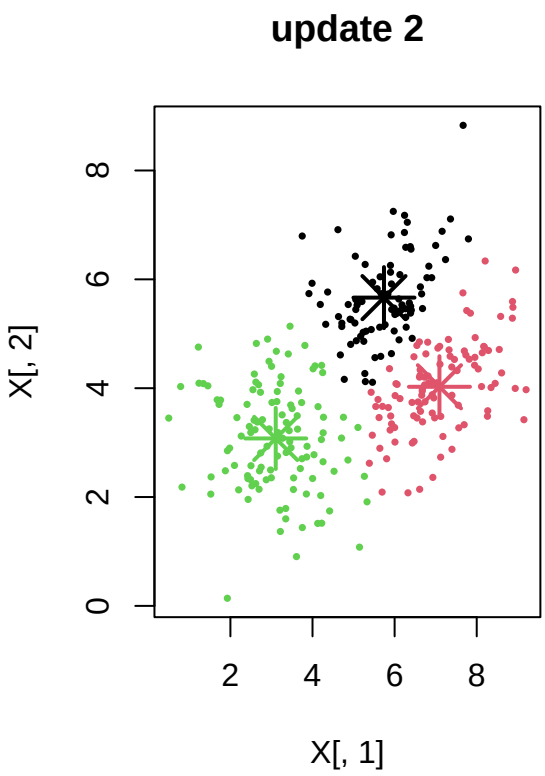

```
## Warning: did not converge in 1 iteration
```

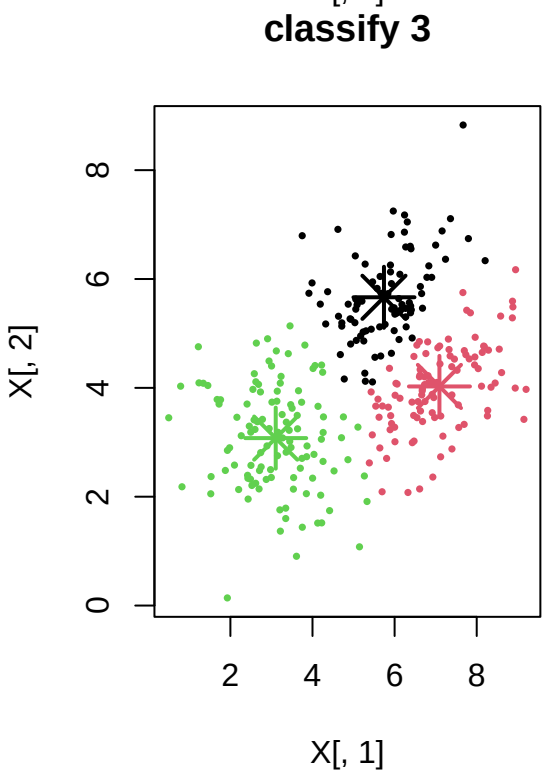


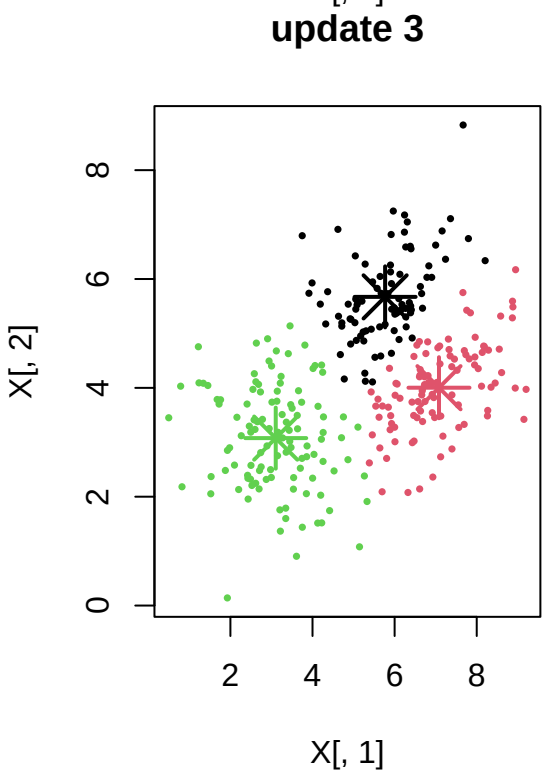

```
## Warning: did not converge in 1 iteration
```

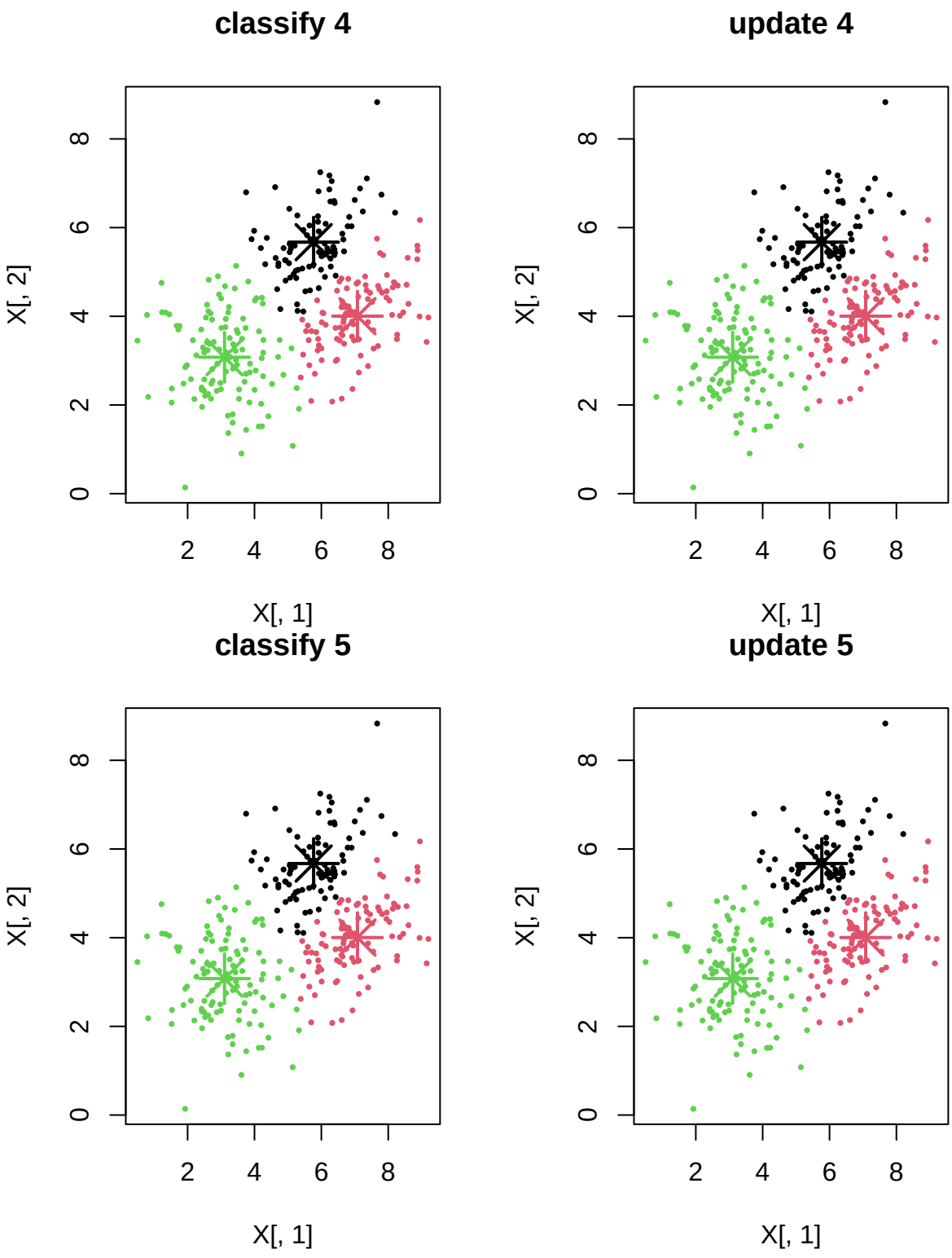



classify 2
X[, 2]
X[, 1]
0
2
4
6
8

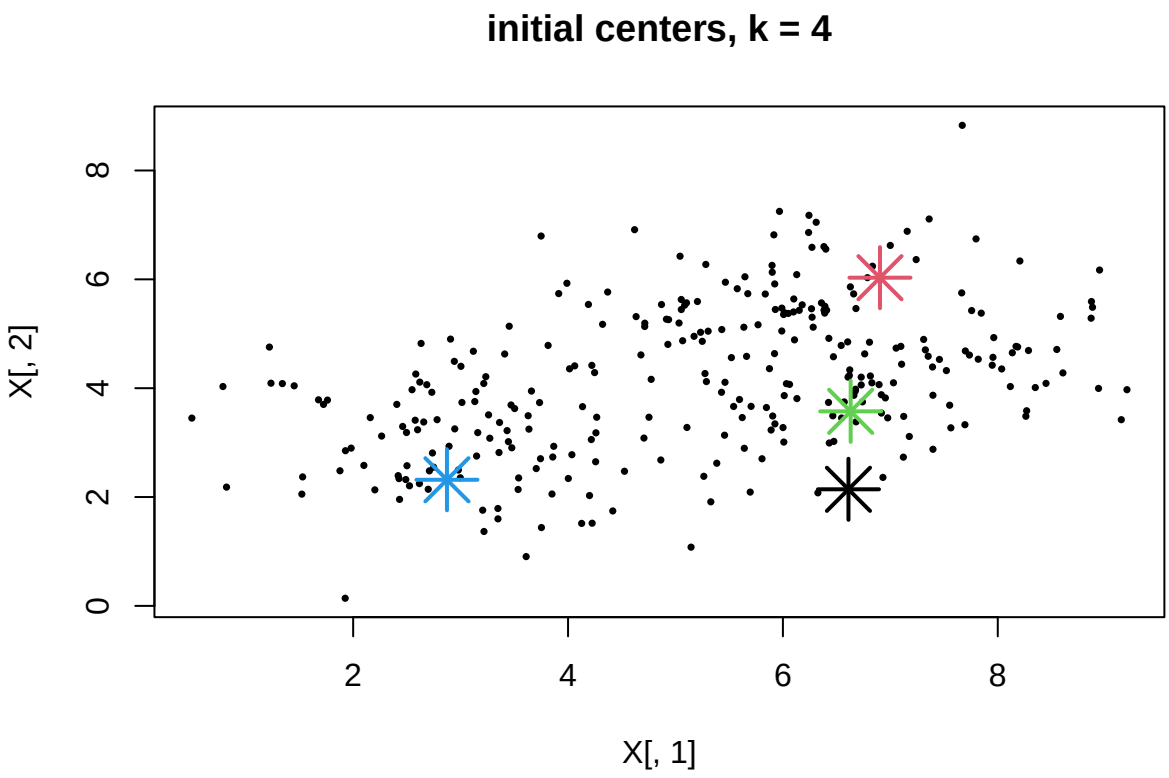

update 2
X[, 2]
X[, 1]
0
2
4
6
8


classify 3
X[, 2]
X[, 1]
0
2
4
6
8

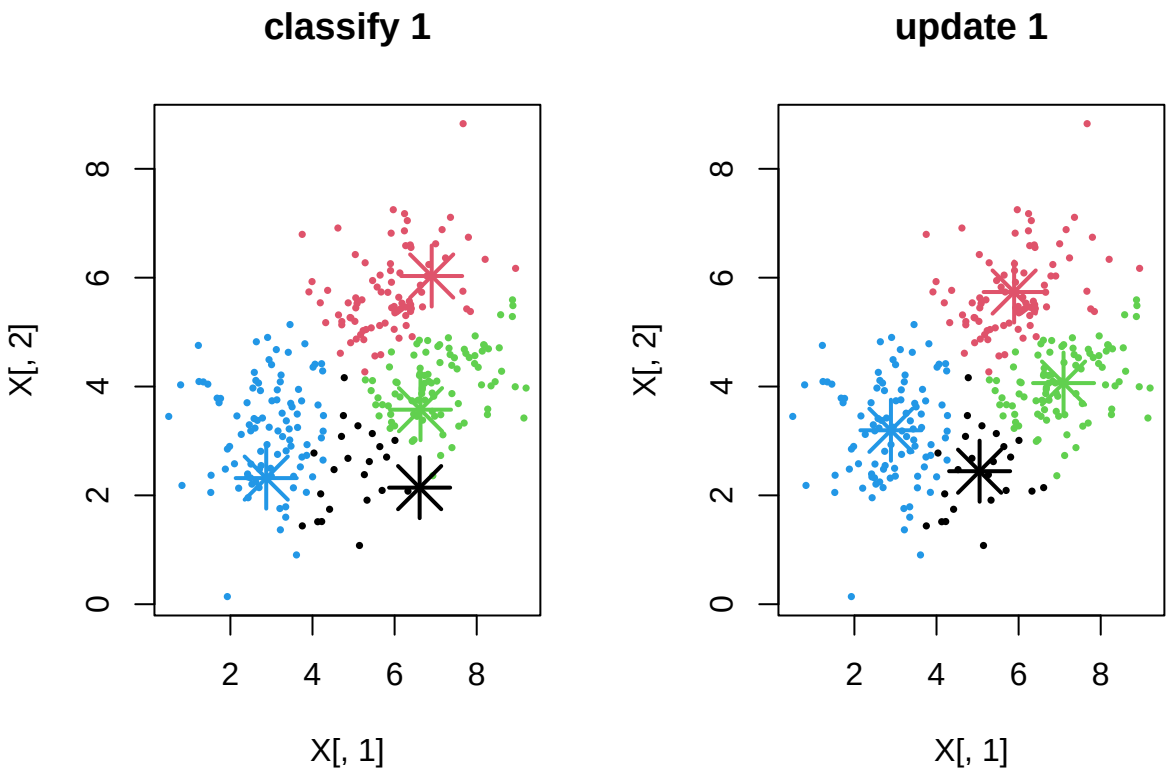

update 3
X[, 2]
X[, 1]
0
2
4
6
8



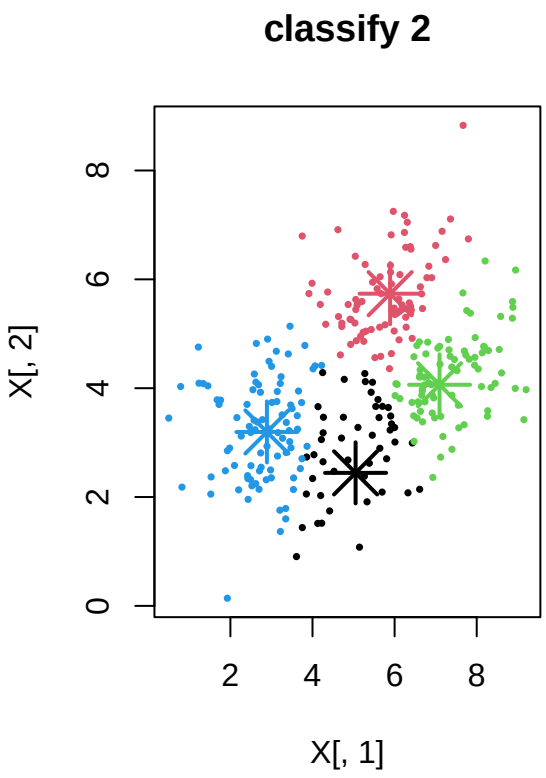

```
## Warning: did not converge in 1 iteration
```

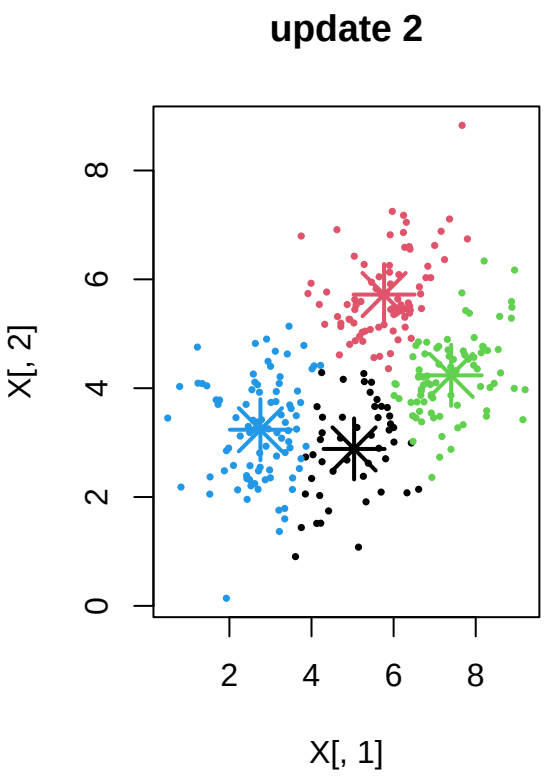



```
## Warning: did not converge in 1 iteration
```

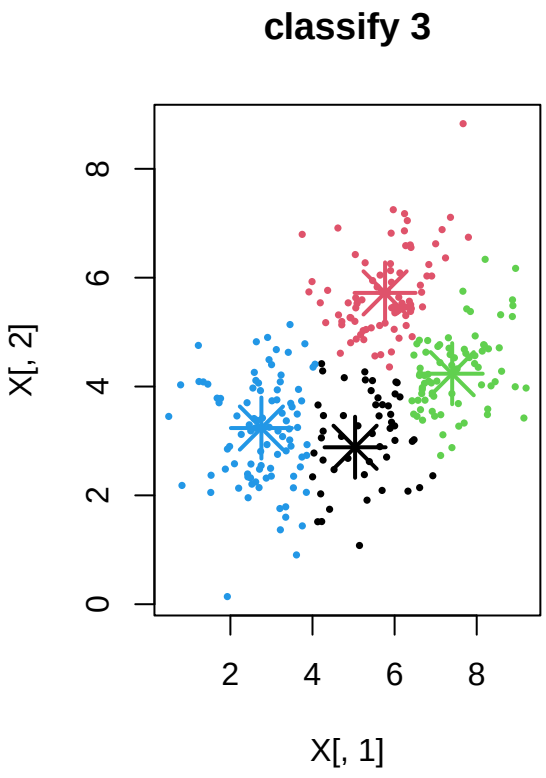


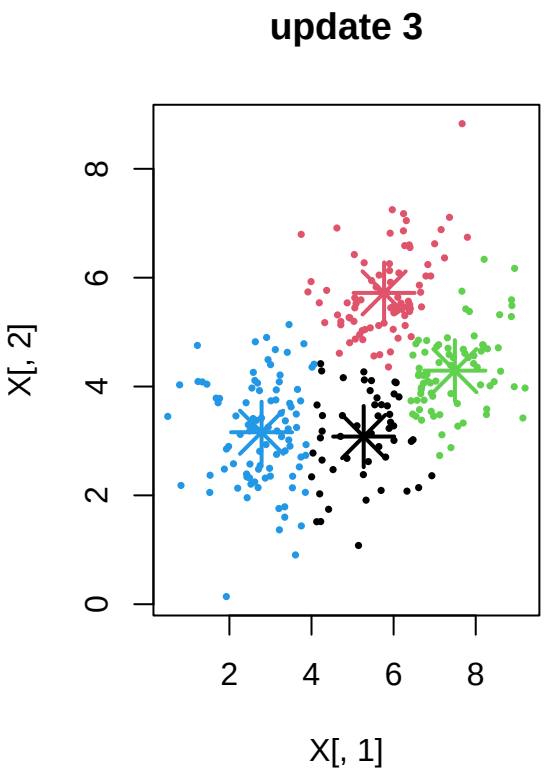

```
## Warning: did not converge in 1 iteration
```

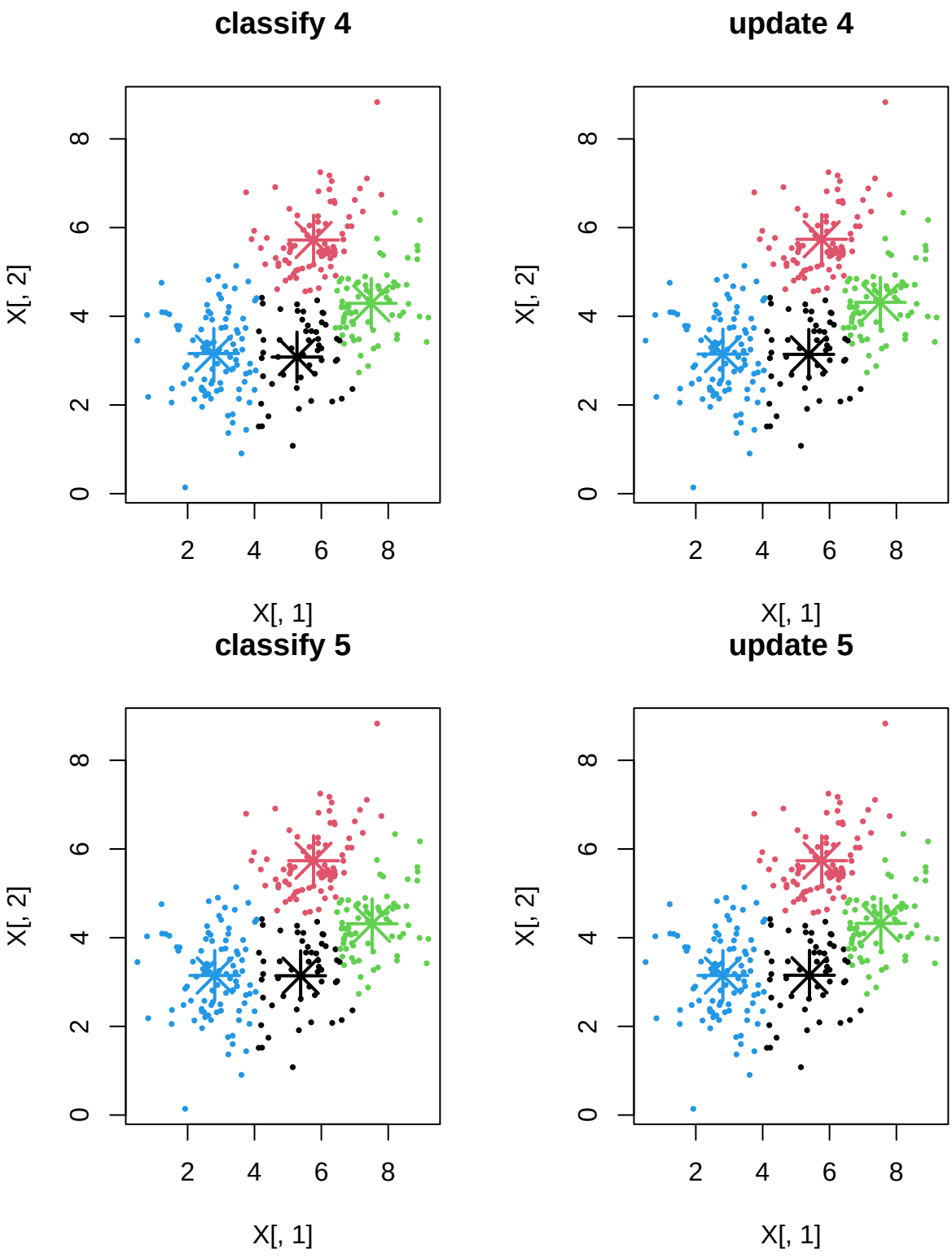




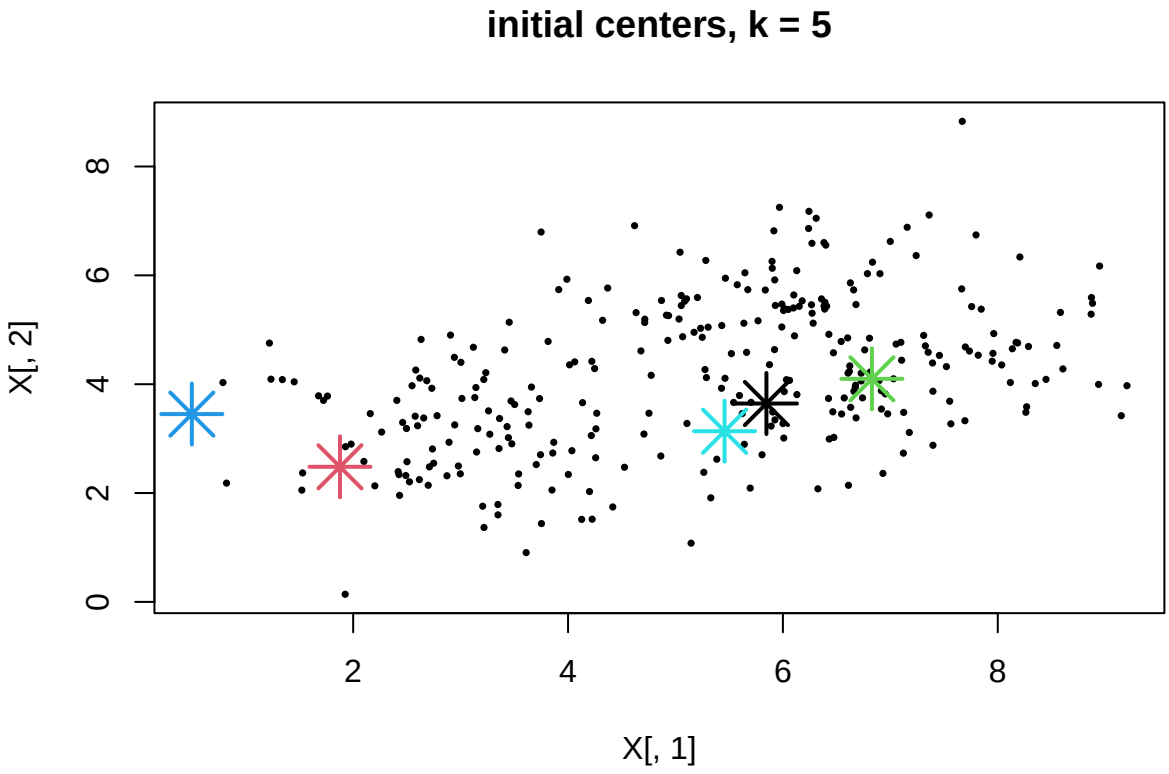

```
## Warning: did not converge in 1 iteration
```



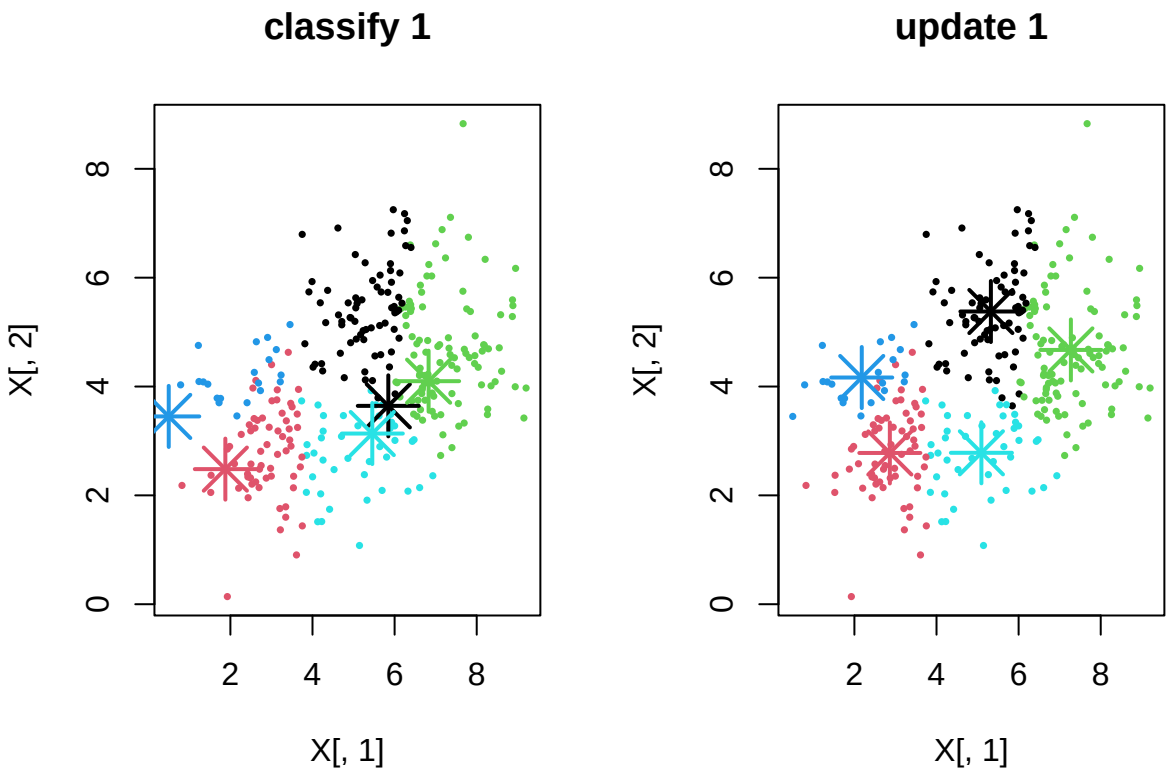




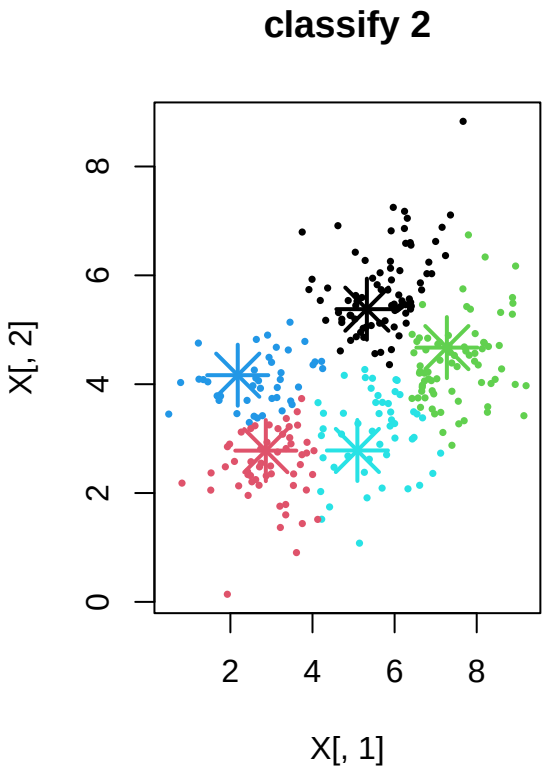

```
## Warning: did not converge in 1 iteration
```

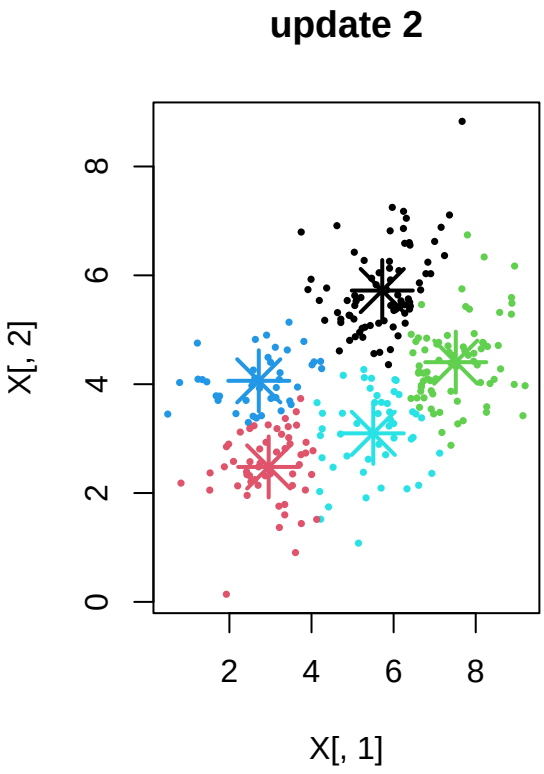



```
## Warning: did not converge in 1 iteration
```

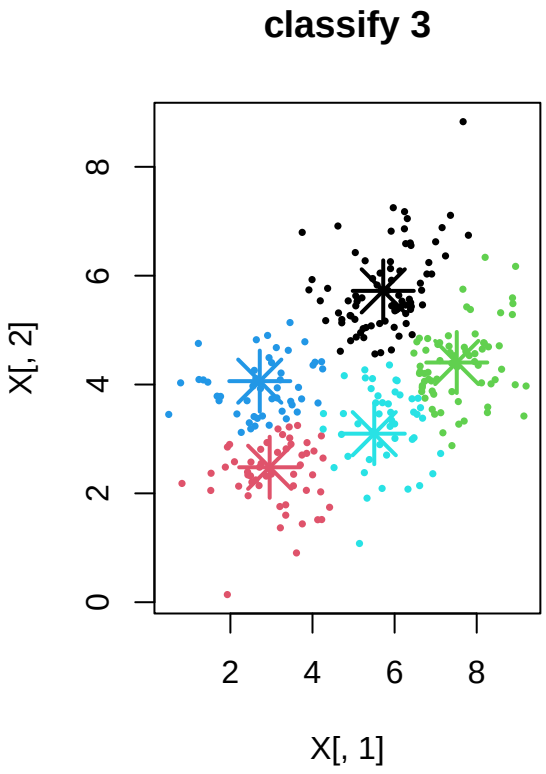


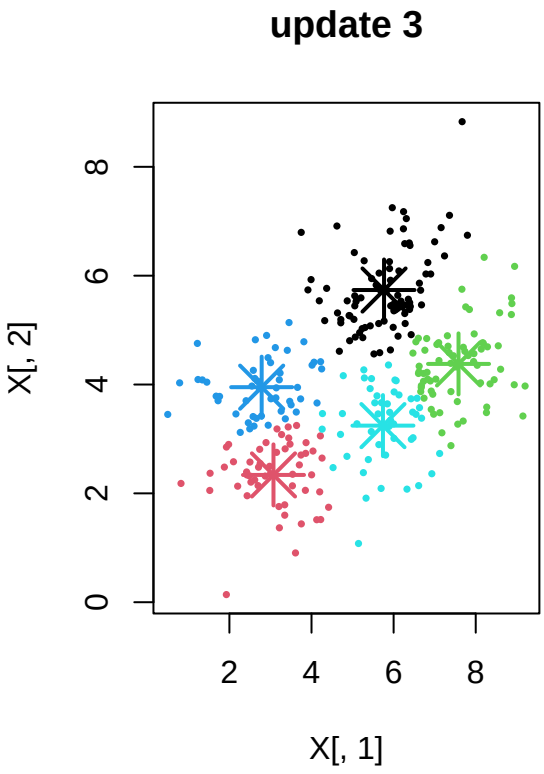

```
## Warning: did not converge in 1 iteration
```

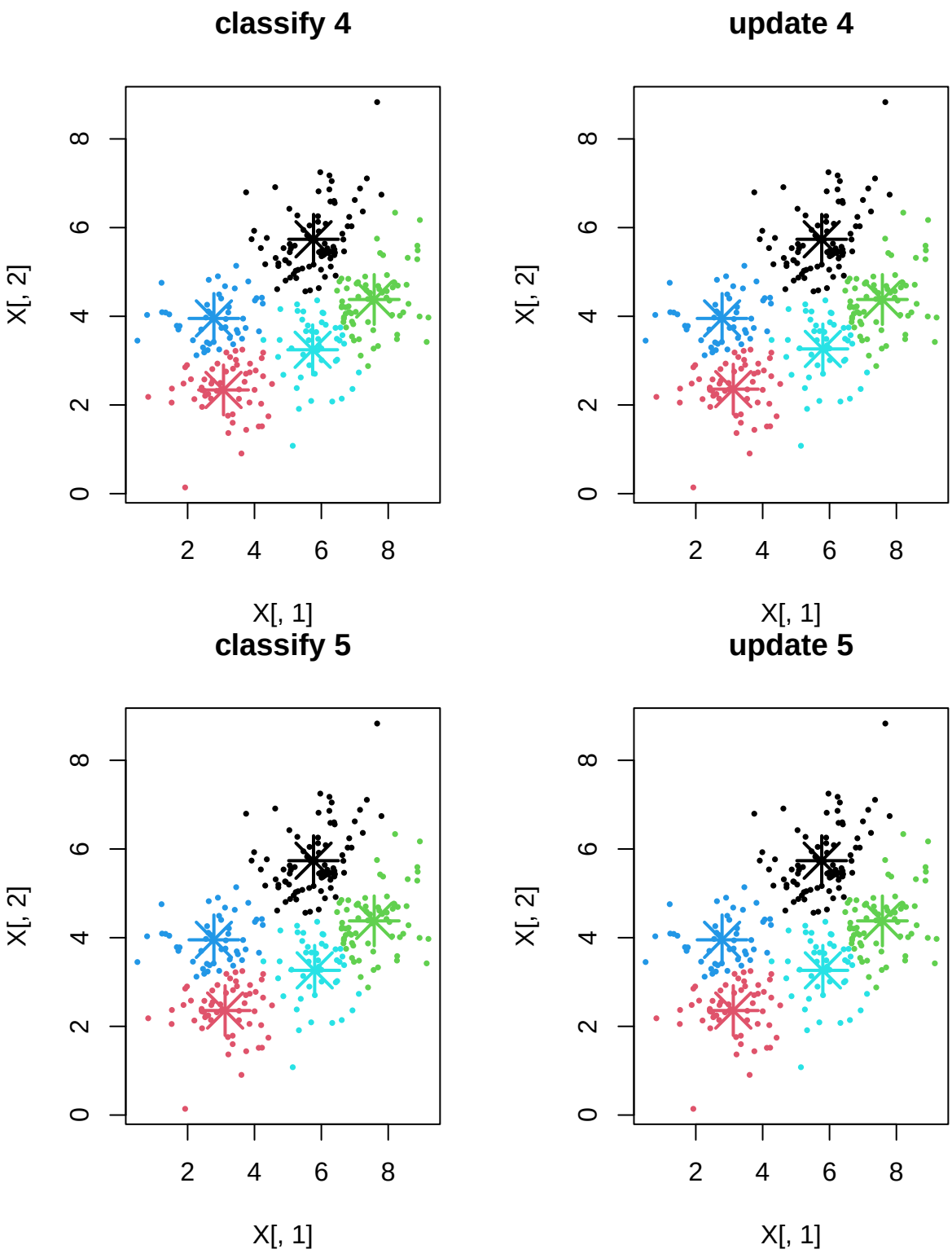





```
## Warning: did not converge in 1 iteration
```





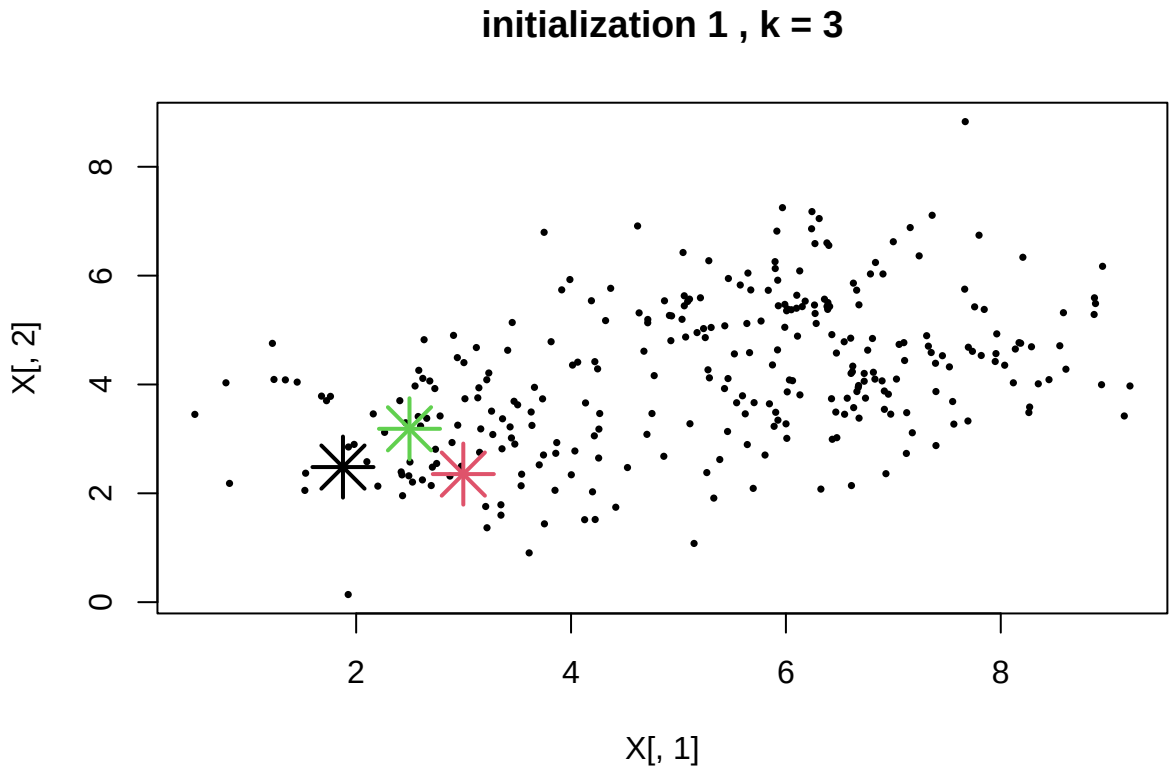


Varying the initialization

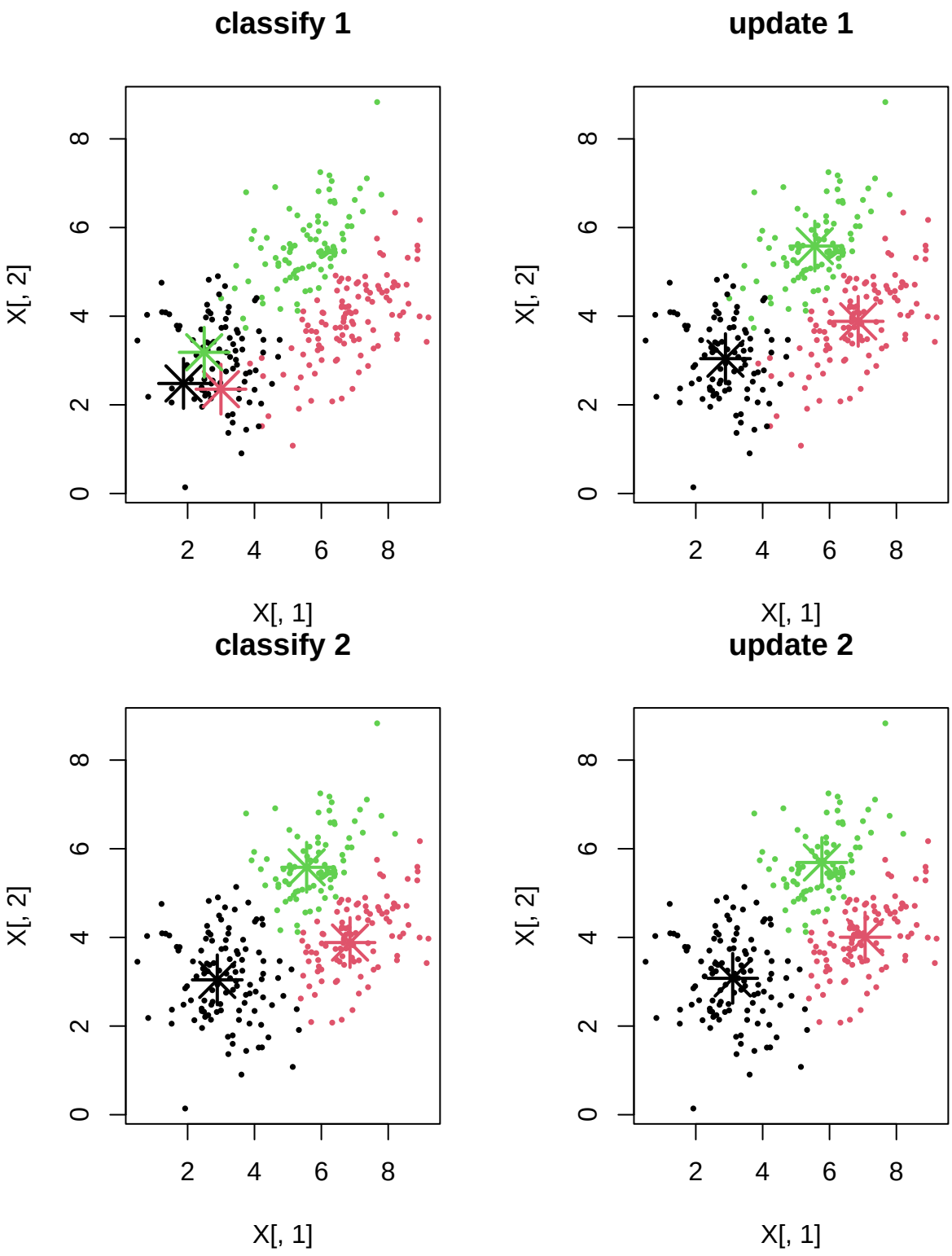
Code to understand K-means algorithm: raw code for k-means

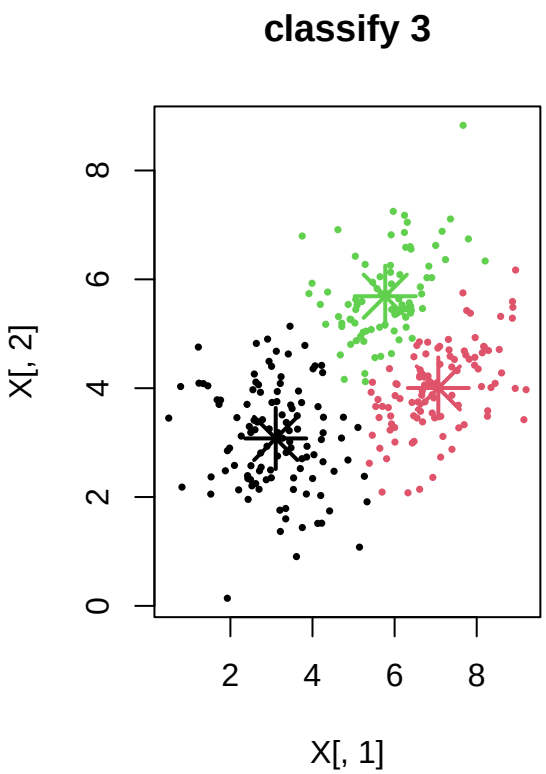
```r
for (k in 3:4) {
  for (i in 1:2) {
    n = nrow(X)
    cens = X[sample(1:n,k),]
    par(mfrow=c(1,1))
    plot(X[,1],X[,2],pch=16, cex = 0.5, main=paste("initialization", i,", k =", k))
    points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd = 2)
    par(mfrow=c(1,2))
    for(i in 1:5) {
      oldcen = cens
      km = kmeans(X,centers=cens,iter.max=1,nstart=1,algorithm="MacQueen")
      plot(X[,1],X[,2],col=km$cluster,pch=16, cex = 0.5, main=paste("classify", i))
      points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd= 2)
      cens = km$centers
      plot(X[,1],X[,2],col=km$cluster,pch=16,cex=0.5, main=paste("update", i))
      points(cens[,1],cens[,2],col=1:k,pch=8,cex=3, lwd= 2)
      ind = sum(diag((oldcen-cens)%*%t(oldcen-cens)))
    }
  }
}
```

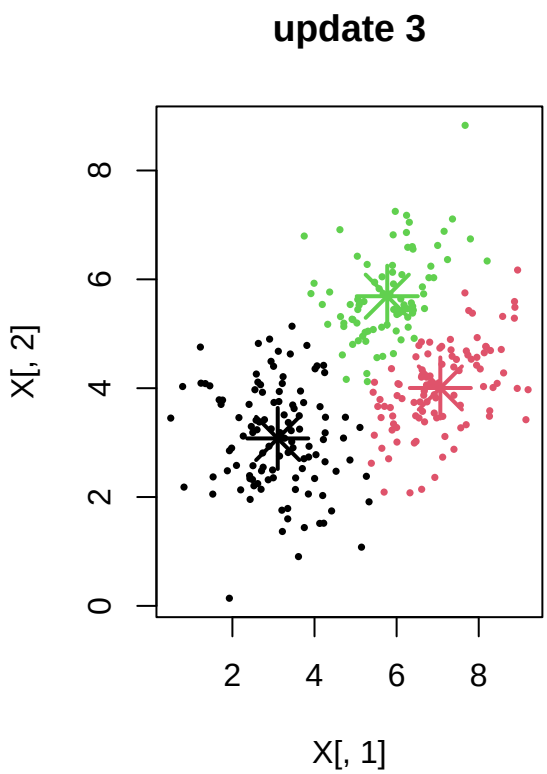
```
## Warning: did not converge in 1 iteration
```

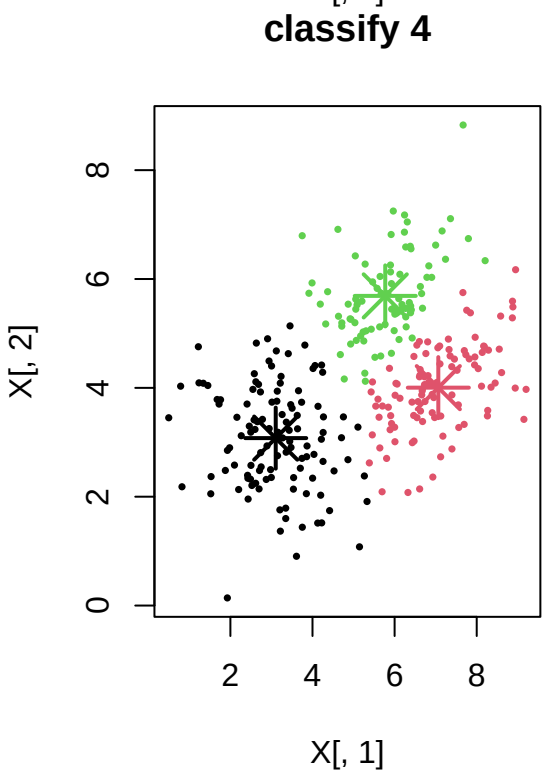


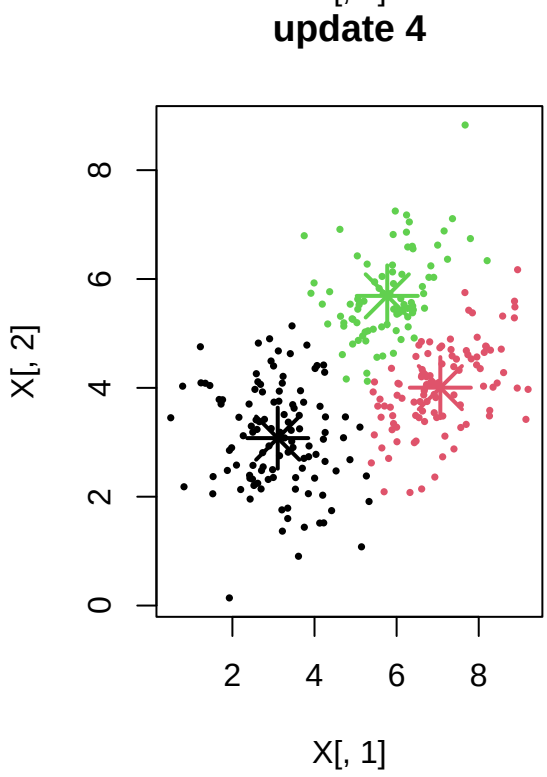

```
## Warning: did not converge in 1 iteration
```

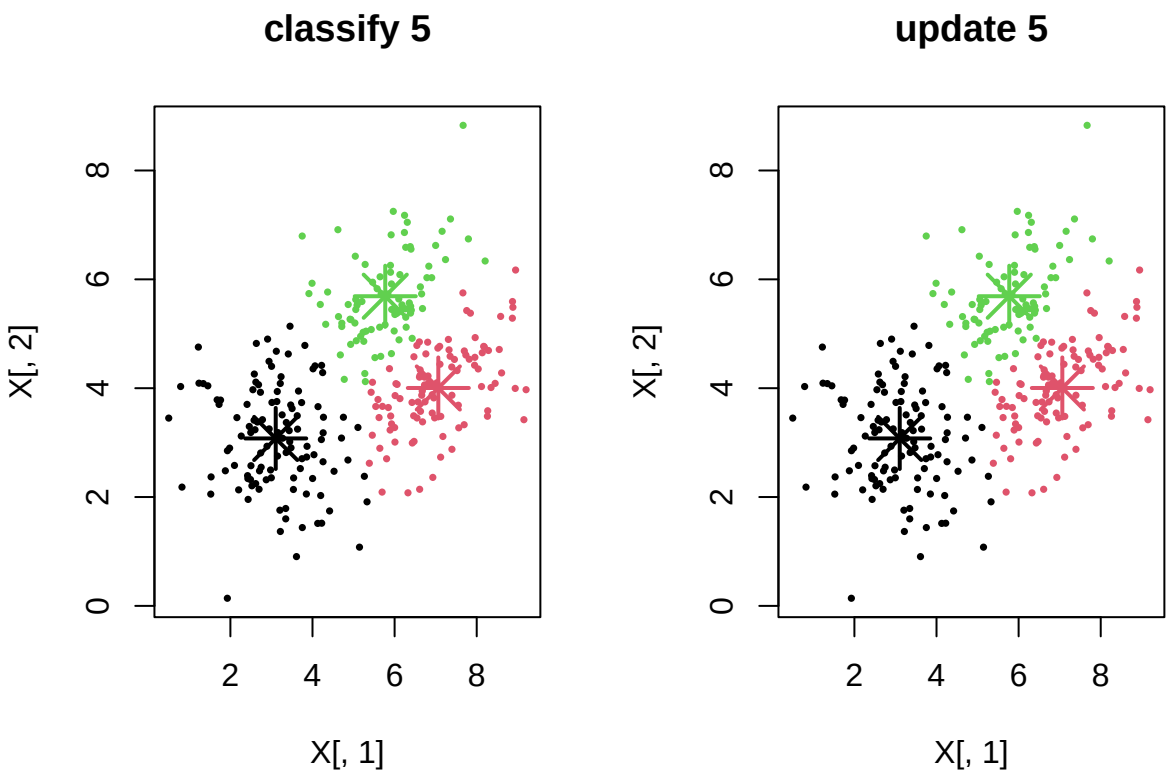

classify 1
update 1
classify 2
update 2
X[, 1]
X[, 2]
0
2
4
6
8


classify 3
X[, 2]
X[, 1]
0
2
4
6
8

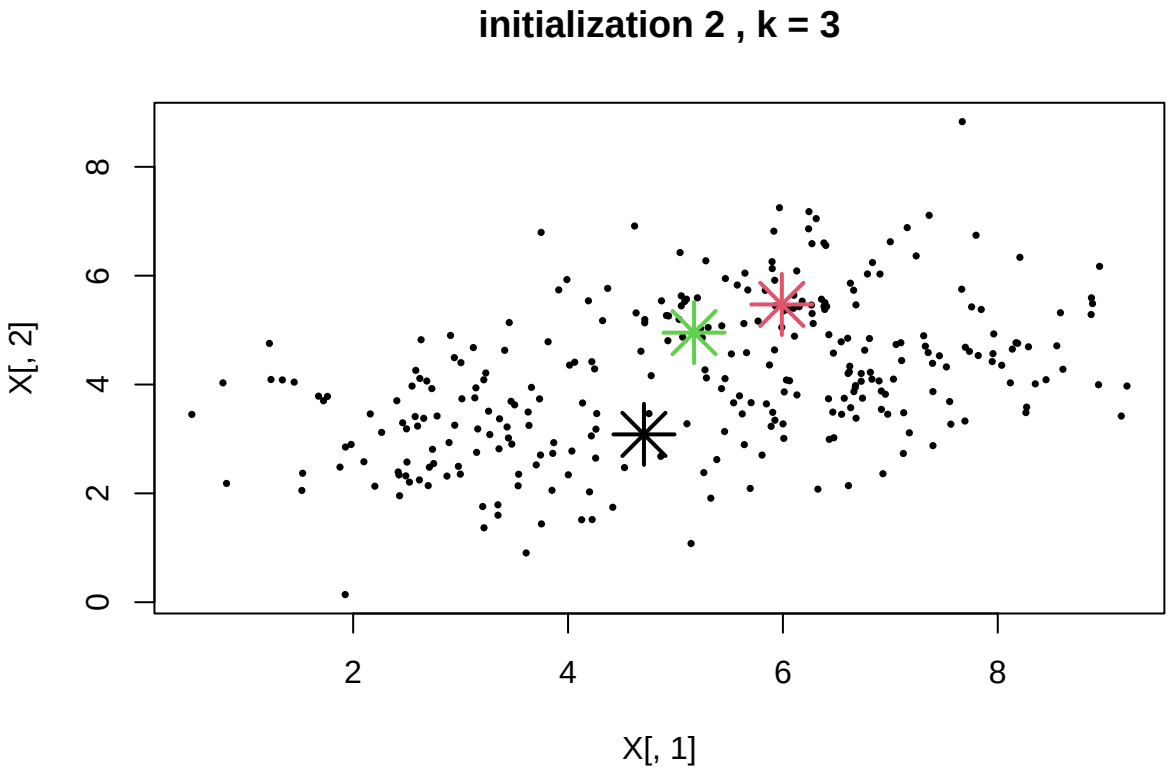

update 3
X[, 2]
X[, 1]
0
2
4
6
8


classify 4
X[, 2]
X[, 1]
0
2
4
6
8

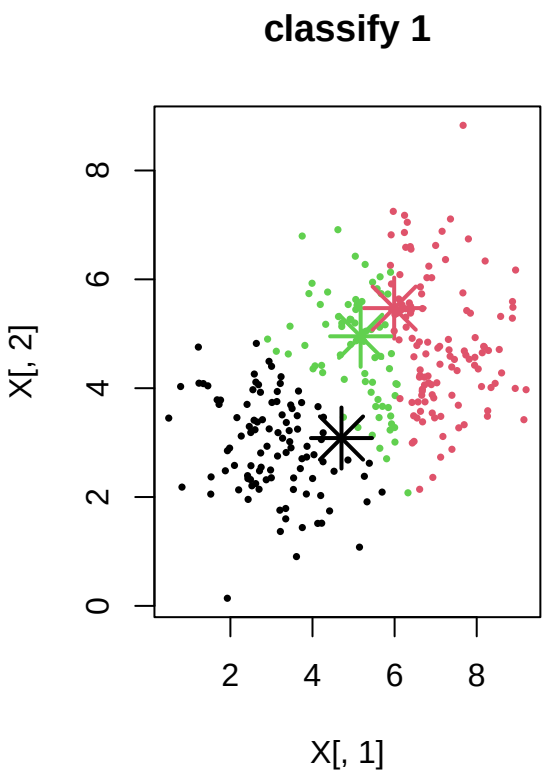

update 4
X[, 2]
X[, 1]
0
2
4
6
8

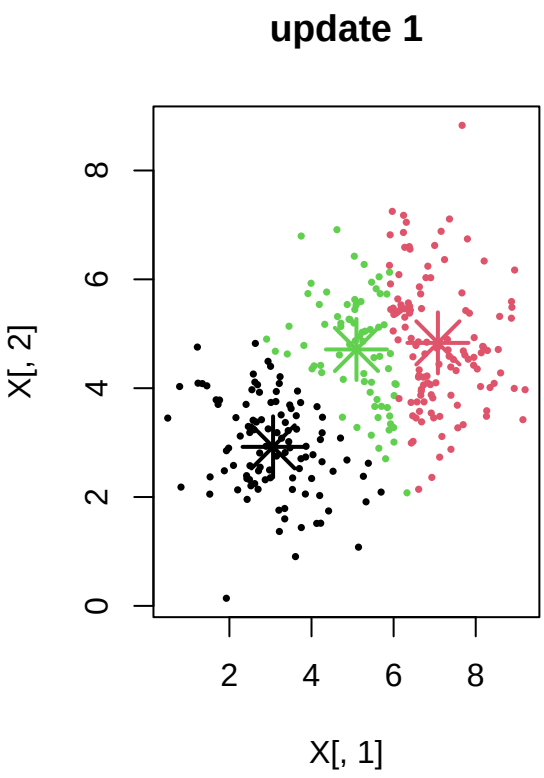



```
## Warning: did not converge in 1 iteration
```

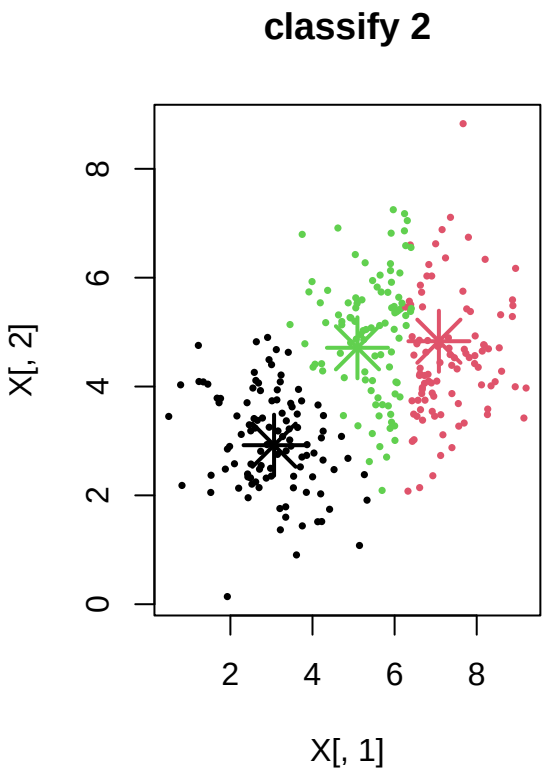


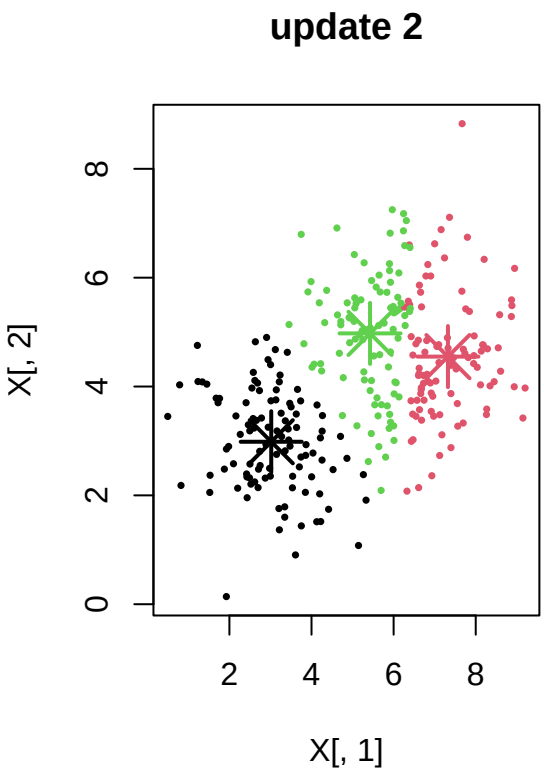

```
## Warning: did not converge in 1 iteration
```



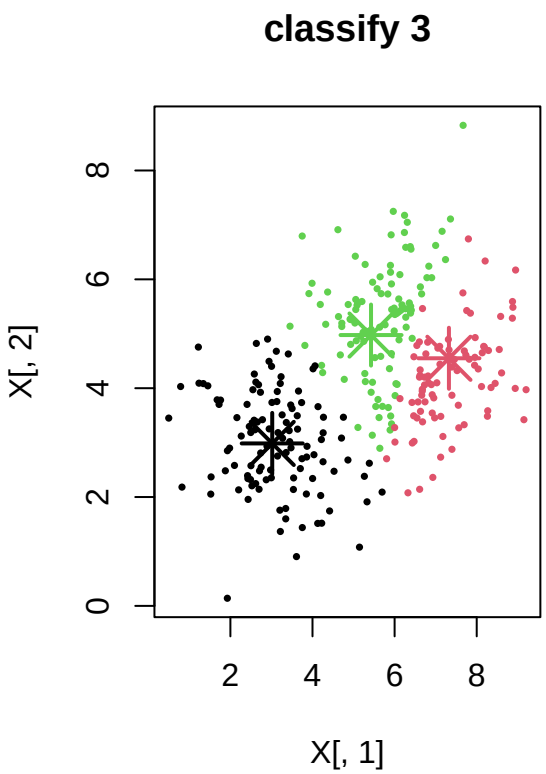


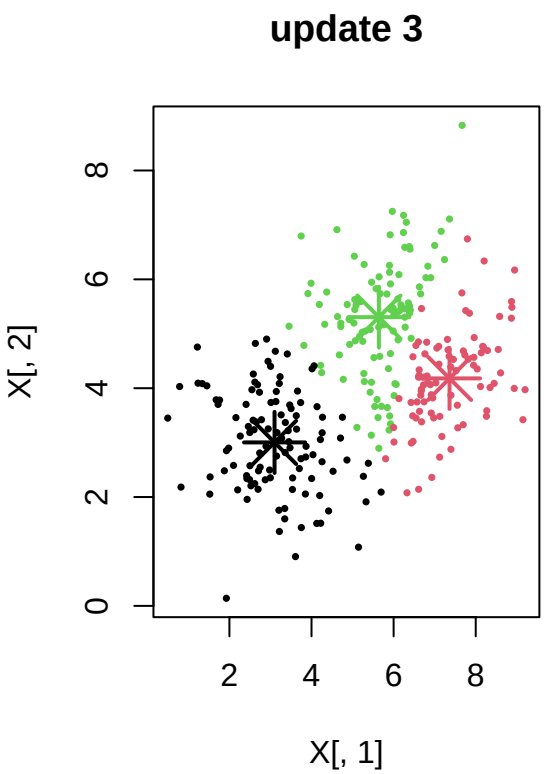

```
## Warning: did not converge in 1 iteration
```

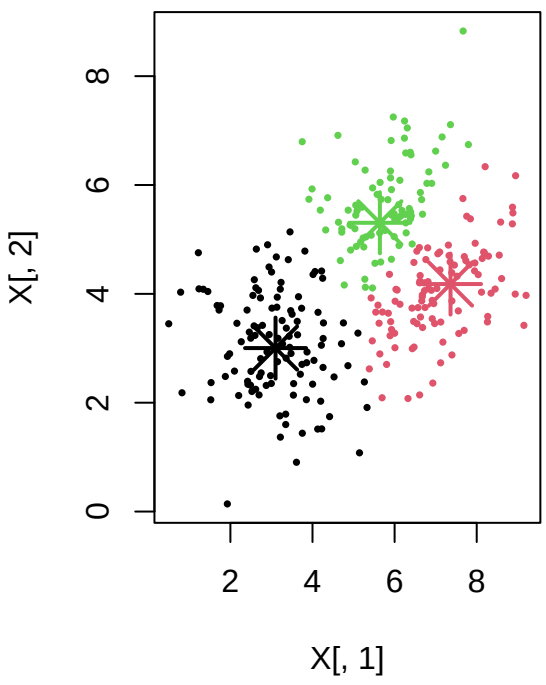


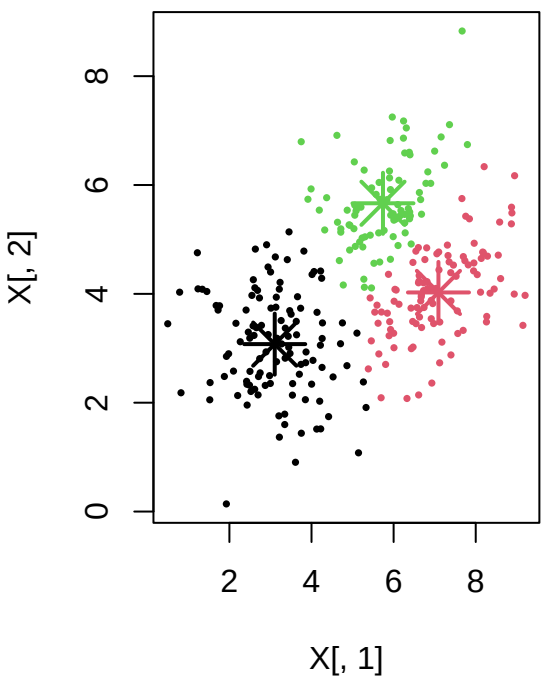


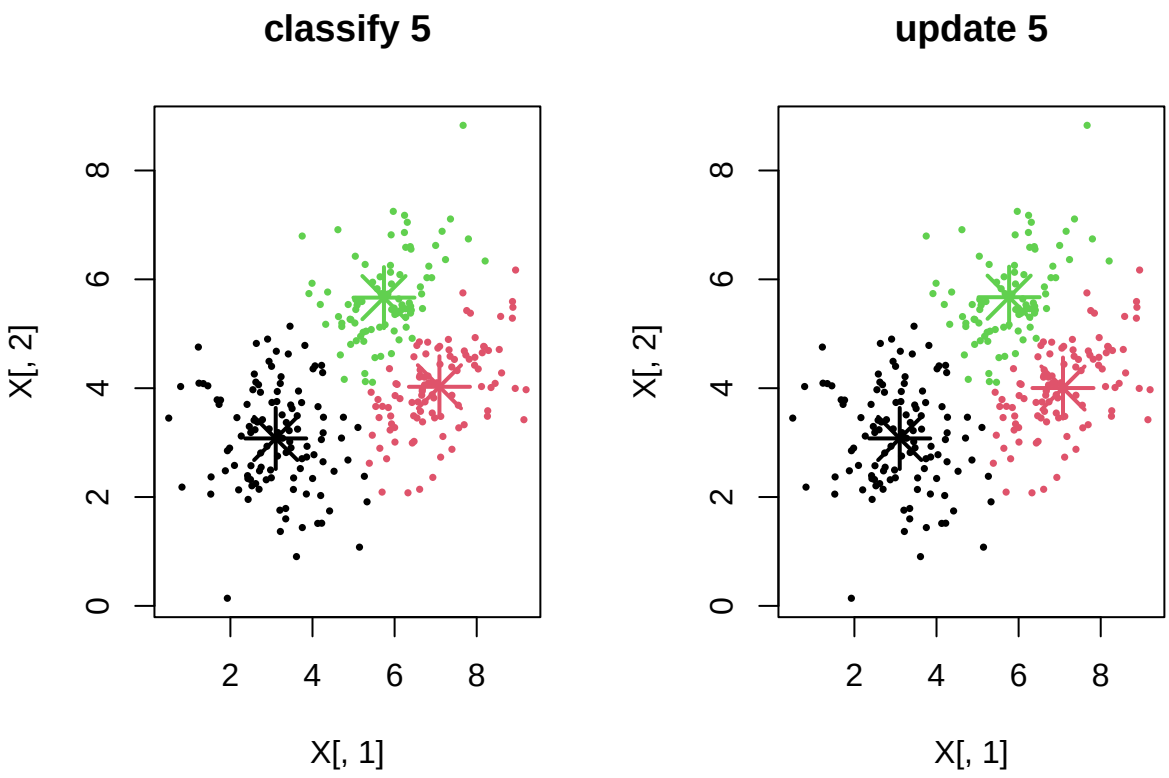

```
## Warning: did not converge in 1 iteration
```


classify 3
X[, 2]
X[, 1]
0
2
4
6
8
2
4
6
8

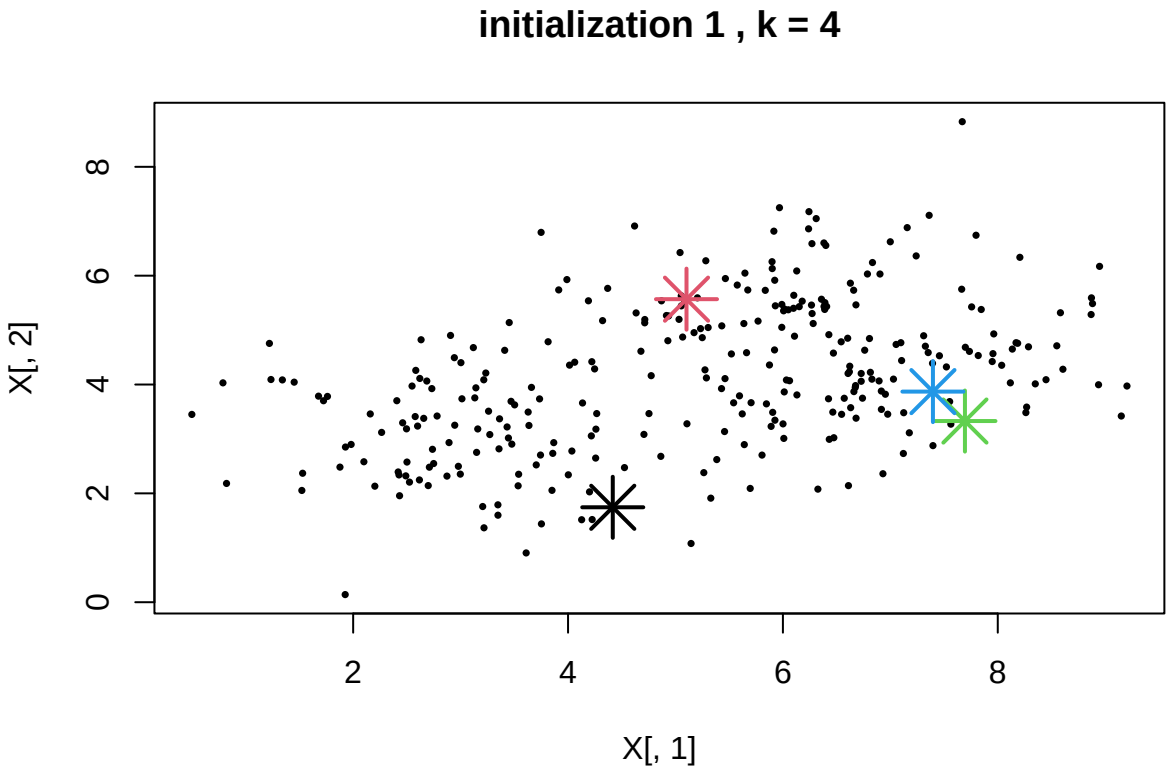

update 3
X[, 2]
X[, 1]
0
2
4
6
8
2
4
6
8


classify 4
X[, 2]
X[, 1]
0
2
4
6
8
2
4
6
8

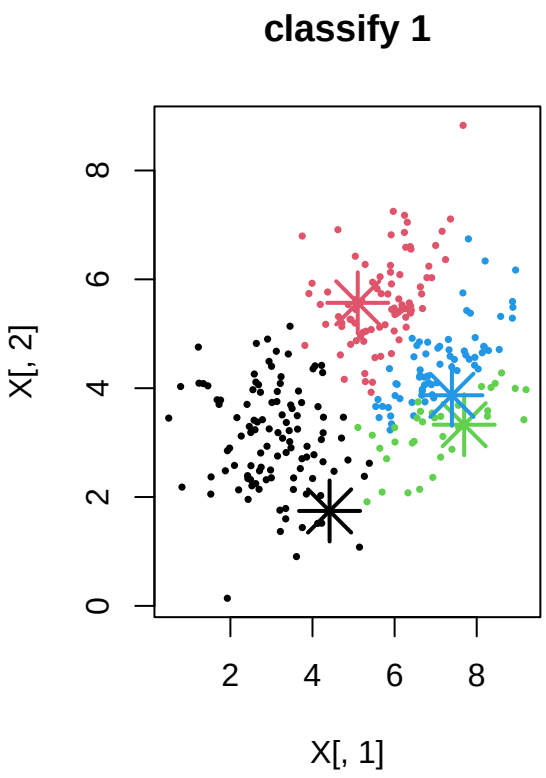

update 4
X[, 2]
X[, 1]
0
2
4
6
8
2
4
6
8

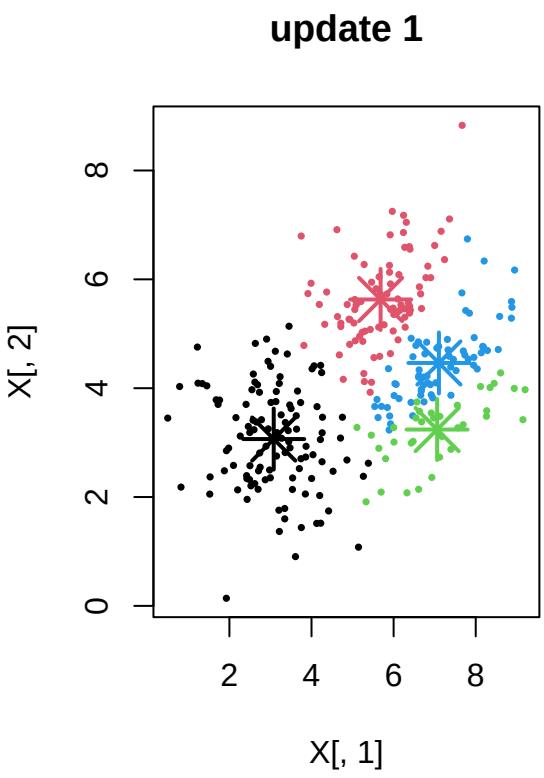



```
## Warning: did not converge in 1 iteration
```

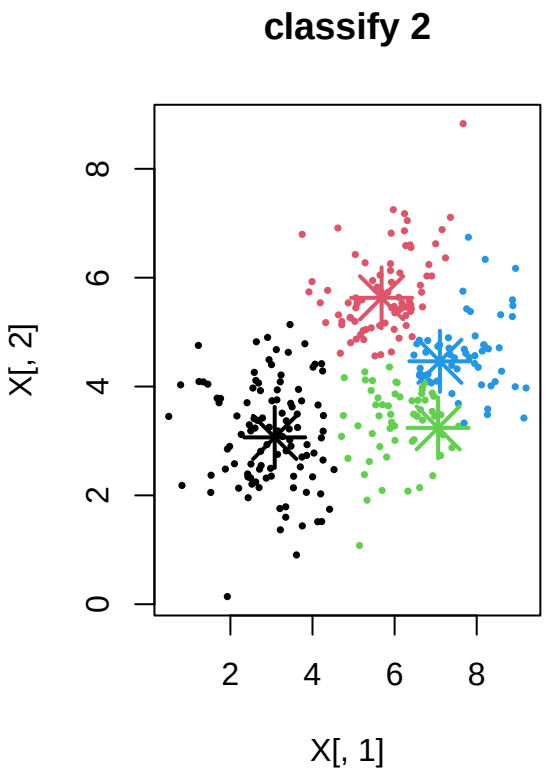


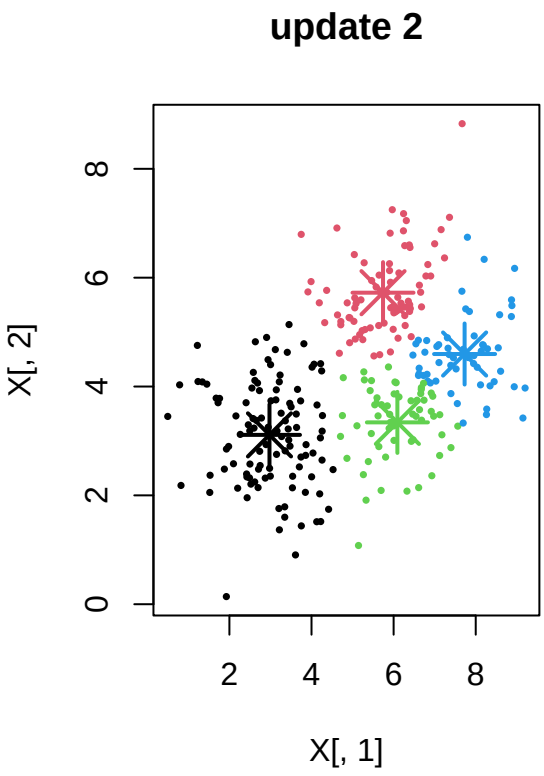

```
## Warning: did not converge in 1 iteration
```

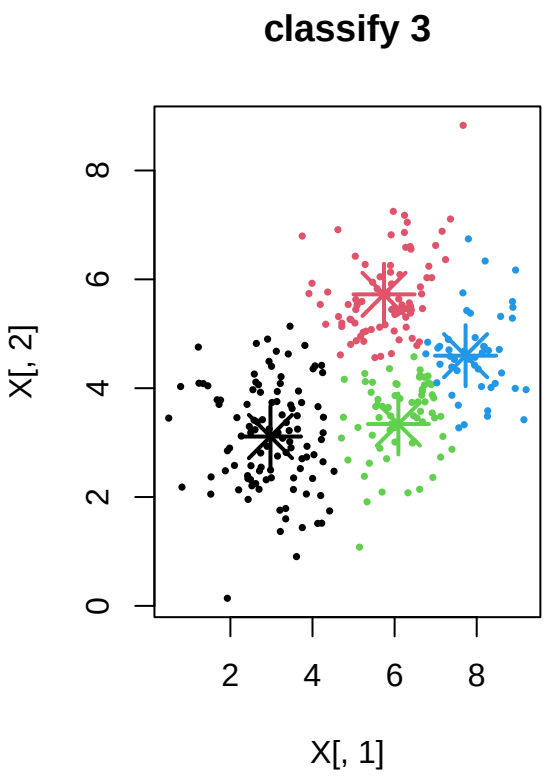


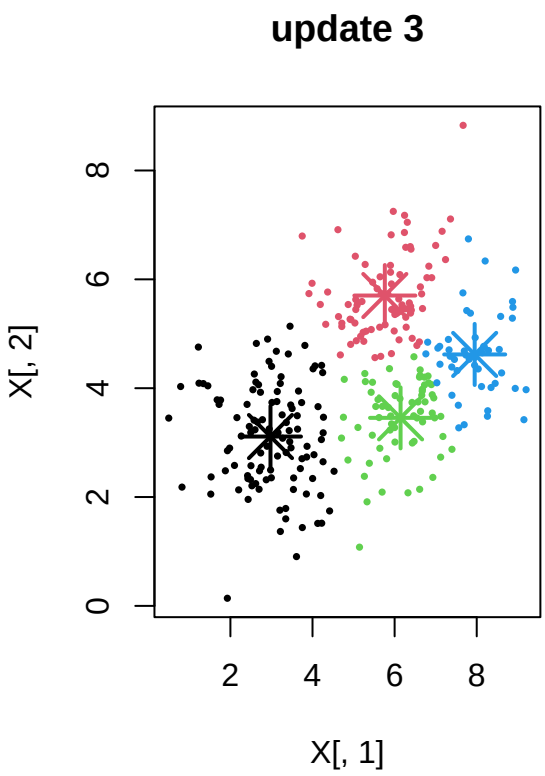


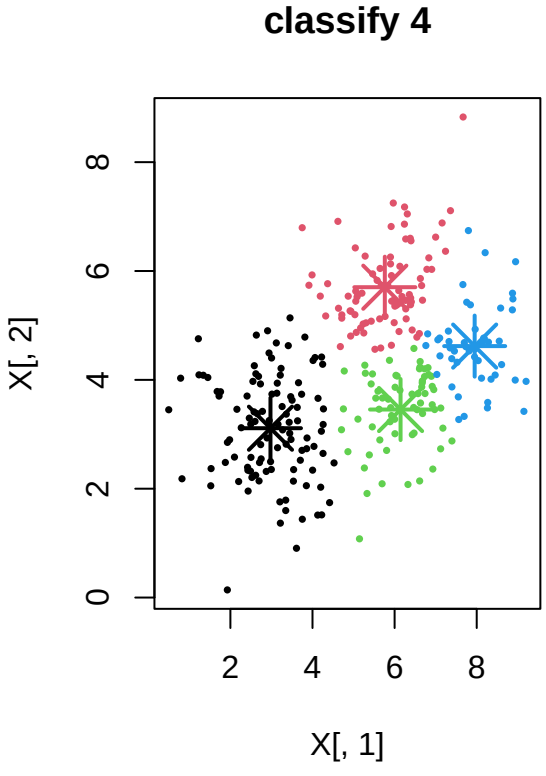

```
## Warning: did not converge in 1 iteration
```

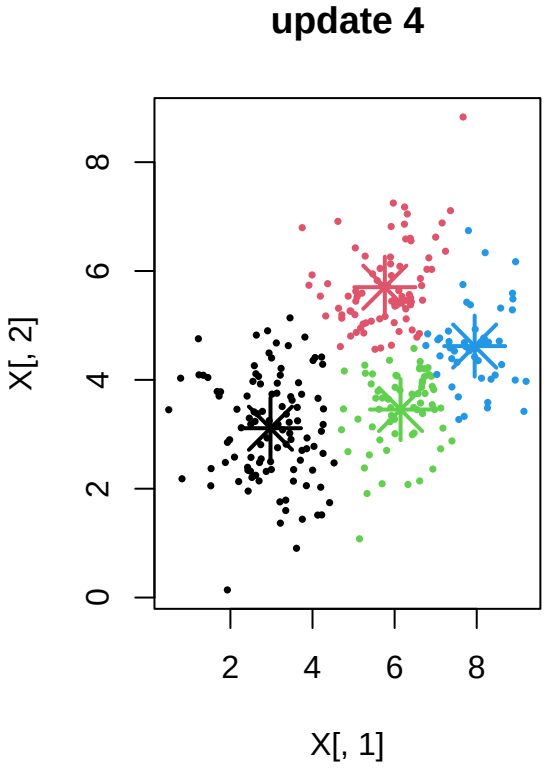


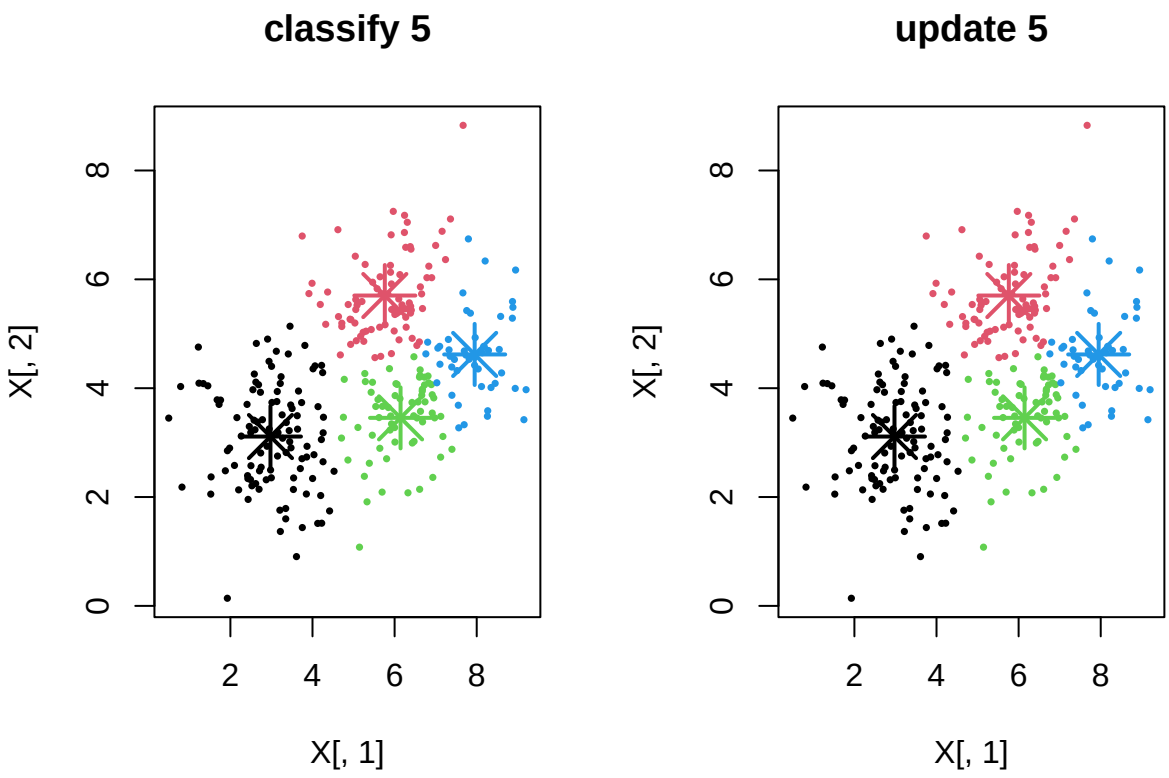



classify 3
X[, 2]
X[, 1]
0
2
4
6
8

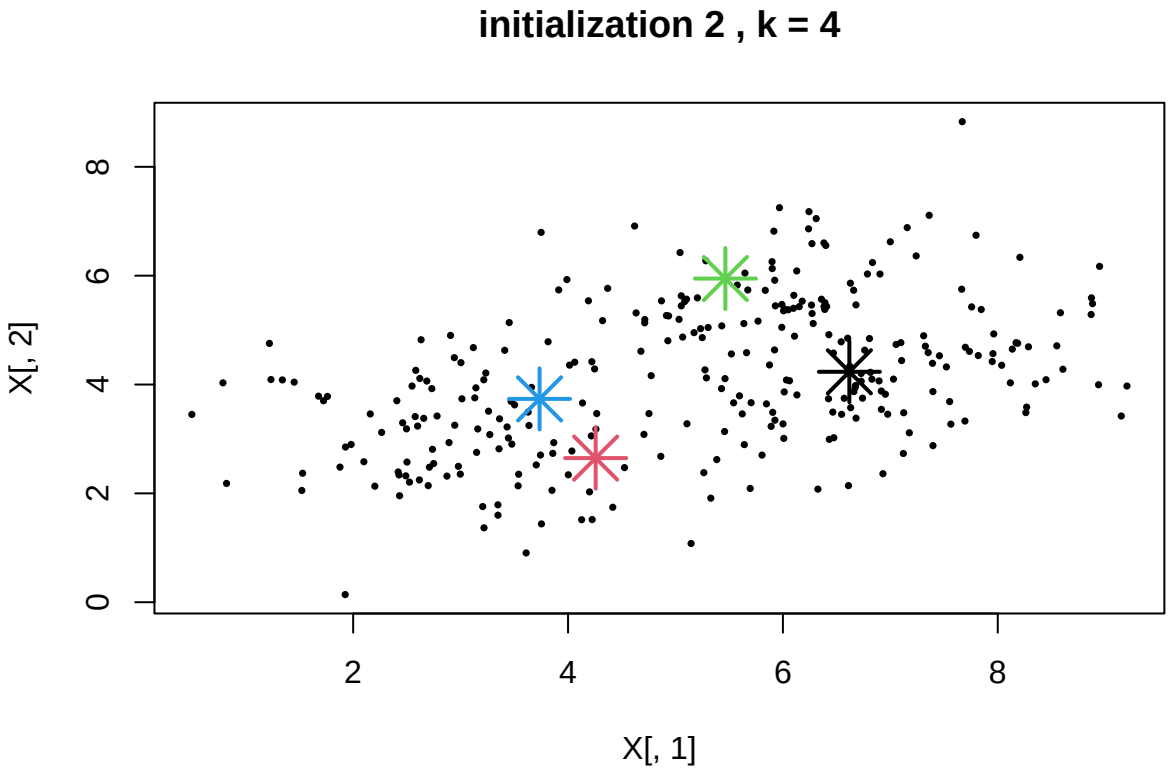

update 3
X[, 2]
X[, 1]
0
2
4
6
8


classify 4
X[, 2]
X[, 1]
0
2
4
6
8

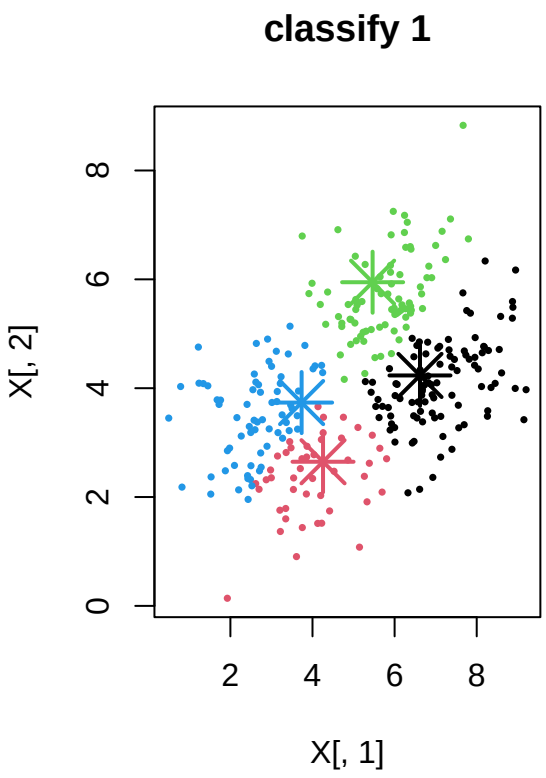

update 4
X[, 2]
X[, 1]
0
2
4
6
8

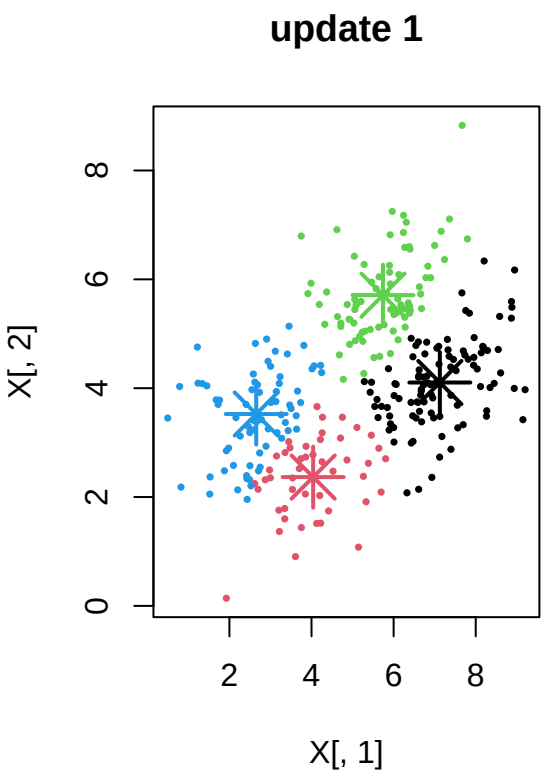



```
## Warning: did not converge in 1 iteration
```

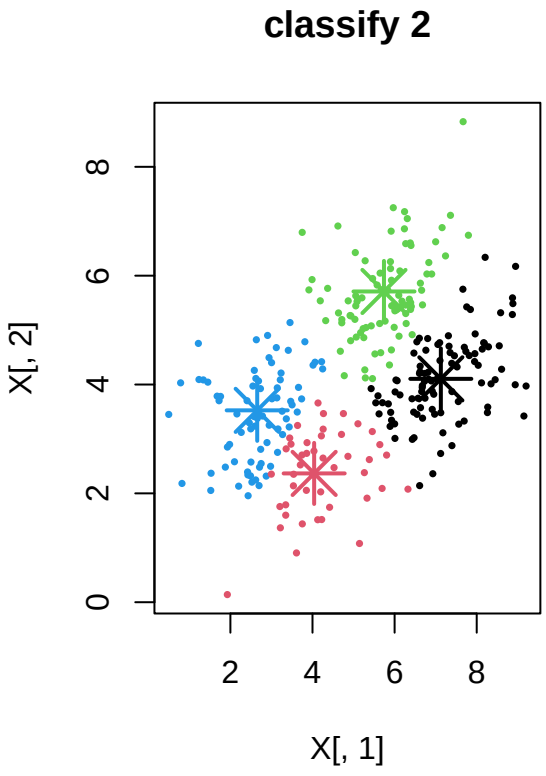


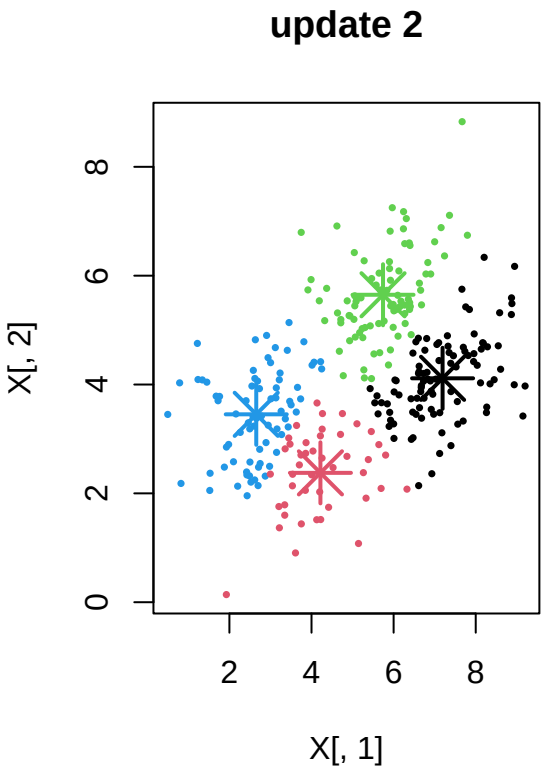

```
## Warning: did not converge in 1 iteration
```

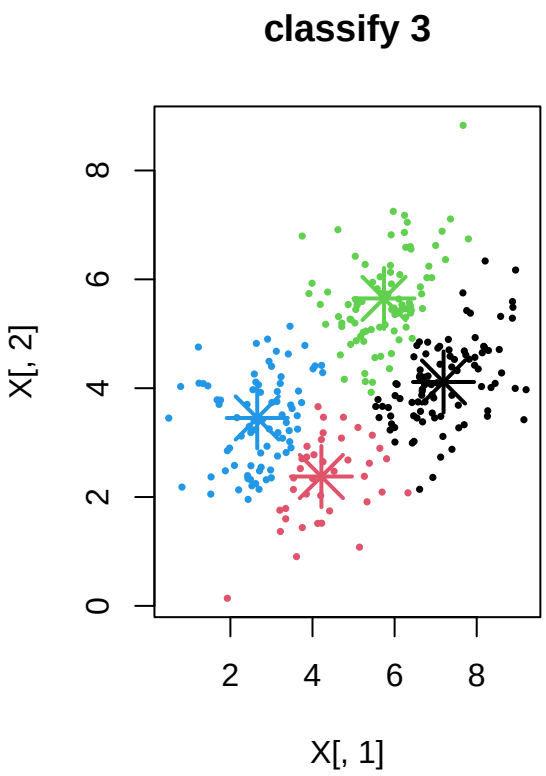


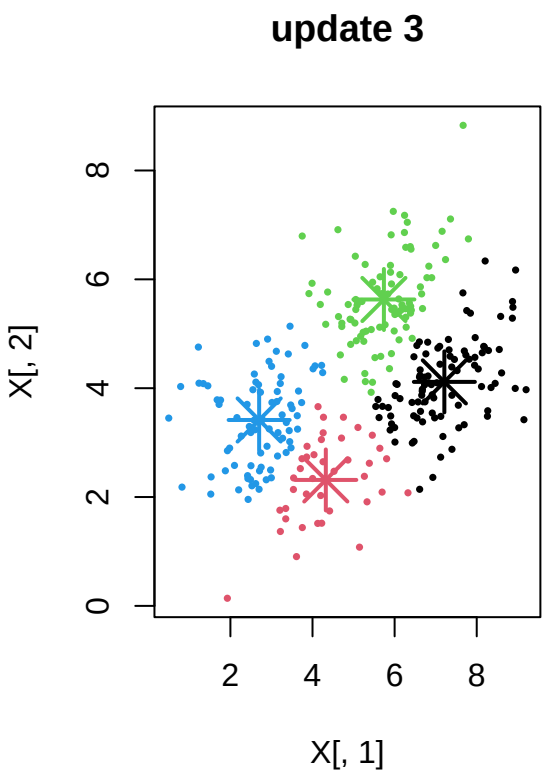


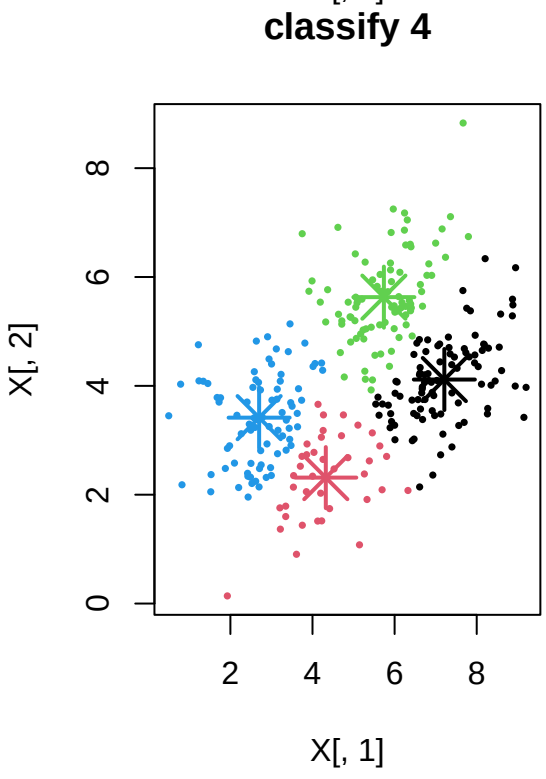

```
## Warning: did not converge in 1 iteration
```

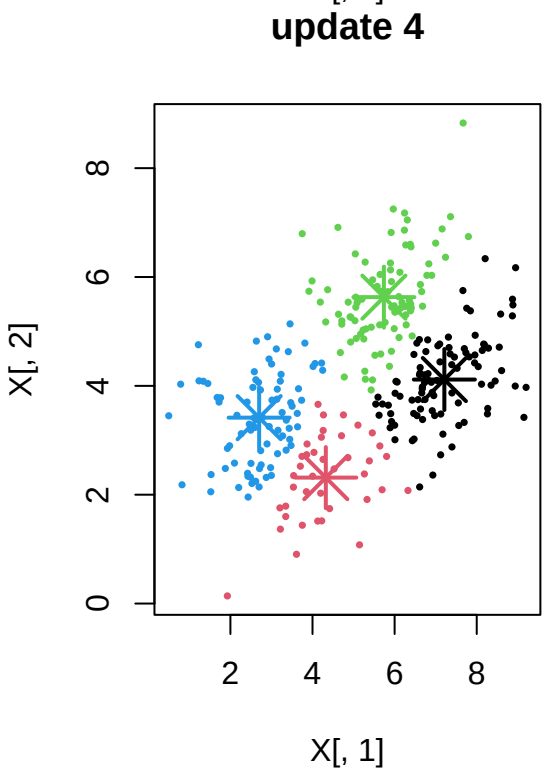


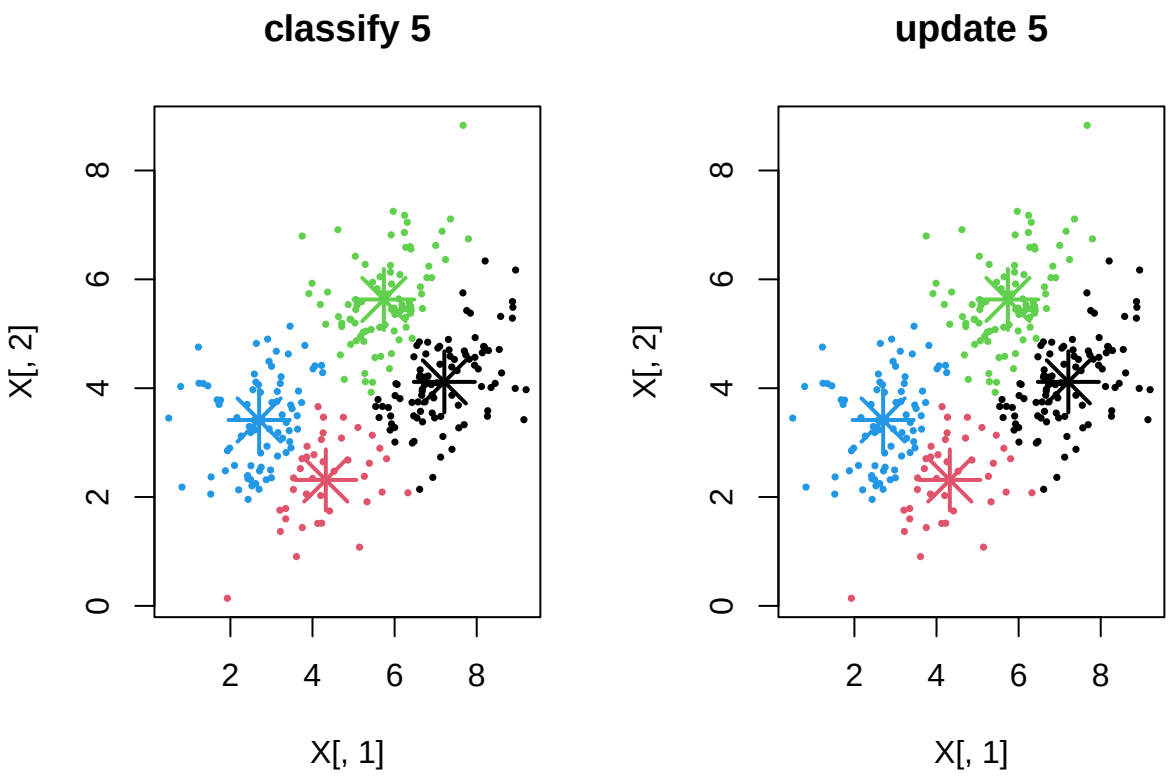



classify 3
X[, 2]
X[, 1]
0
2
4
6
8


update 3
X[, 2]
X[, 1]
0
2
4
6
8


classify 4
X[, 2]
X[, 1]
0
2
4
6
8


update 4
X[, 2]
X[, 1]
0
2
4
6
8




```
load("data/chu.rdata")

chu.x = t(chu$sc_cnt)
chu.y = chu$sc_label
chu.colors = as.integer(factor(chu.y))
```

Compute K-means, then visualize using PCA

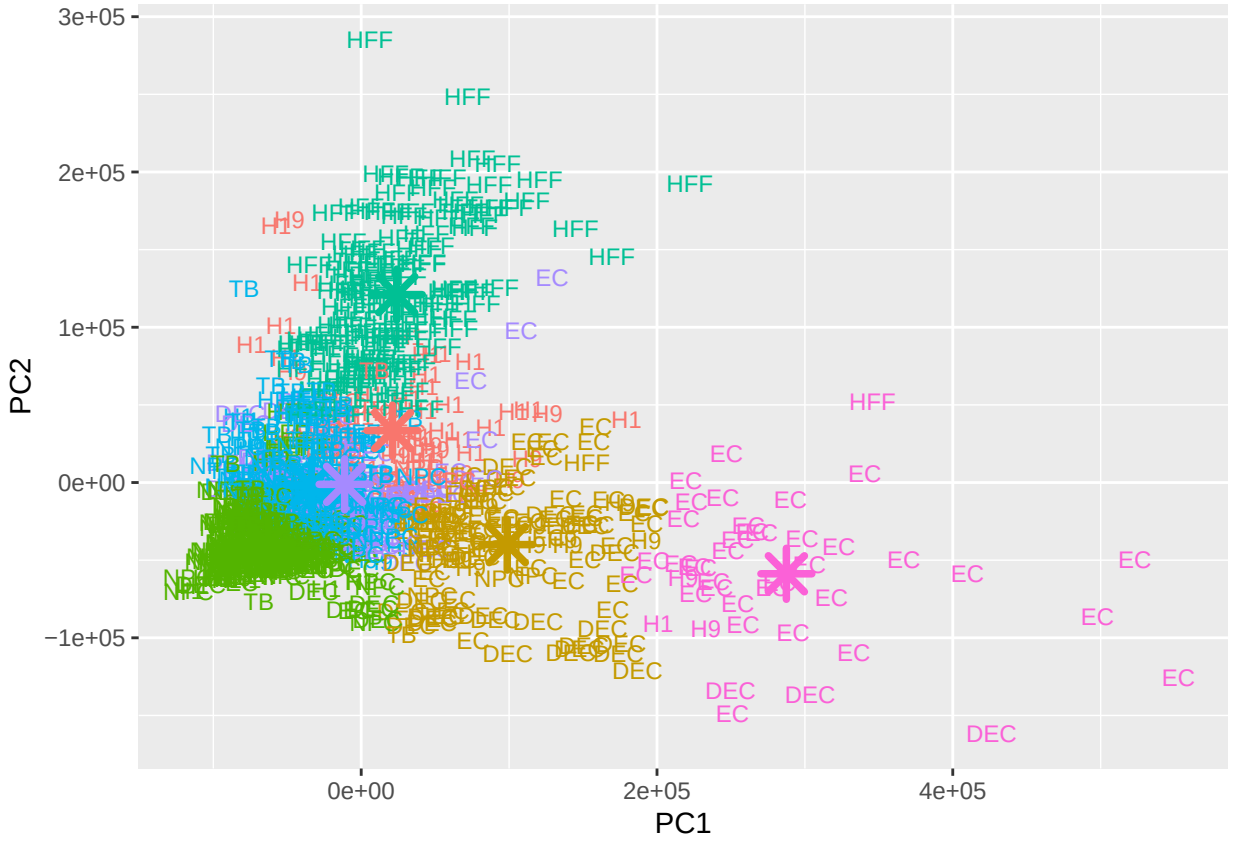
How do we visualize K-means results?
PCA - take SVD to get solution

```
X = scale(chu.x,center=TRUE,scale=FALSE)
sv = svd(X)
U = sv$u
V = sv$v
D = sv$d
Z = X%*%V

K = 7
km = kmeans(X,centers=K)
```

Visualization

```
clustered = data.frame(cbind(Z[,1],Z[,2],km$cluster,chu.y),stringsAsFactors = FALSE)
colnames(clustered) = c("PC1","PC2","PredLabel","CellTuning")
clustered$PC1 = as.numeric(clustered$PC1)
clustered$PC2 = as.numeric(clustered$PC2)
# projected k-means centers
group.data = data.frame(km$centers%*%V[,1:2])
group.data$label = rownames(group.data)
```

```
colnames(group.data) = c("PC1","PC2","PredLabel")

ggplot(clustered,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CellTuning), size = 3) +
  geom_point(data = group.data,size = 5, shape = 8, stroke = 2) +
  theme(legend.position="none")
```

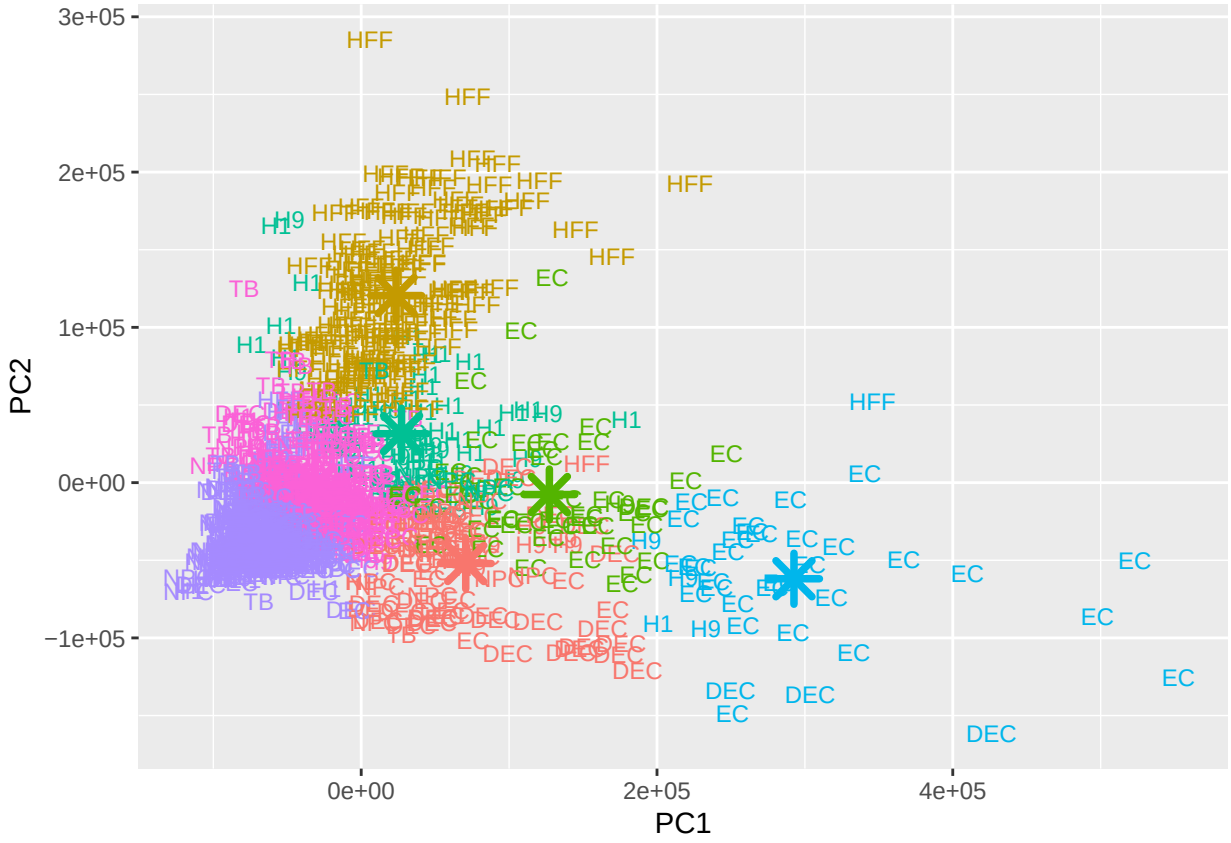


Compute PCA, perform K-means on the largest principal components

```
X = scale(chu.x,center=TRUE,scale=FALSE)
sv = svd(X)
U = sv$u
V = sv$v
D = sv$d
Z = X%*%V


K = 7
km = kmeans(Z,centers=K)
```

```
clustered = data.frame(cbind(Z[,1],Z[,2],km$cluster,chu.y),stringsAsFactors = FALSE)
colnames(clustered) = c("PC1","PC2","PredLabel","CellTuning")
clustered$PC1 = as.numeric(clustered$PC1)
clustered$PC2 = as.numeric(clustered$PC2)
```

```
# projected k-means centers
group.data = data.frame(km$centers[,1:2])
group.data$label = rownames(group.data)
colnames(group.data) = c("PC1","PC2","PredLabel")

ggplot(clustered,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CellTuning), size = 3) +
  geom_point(data = group.data,size = 5, shape = 8, stroke = 2) +
  theme(legend.position="none")
```



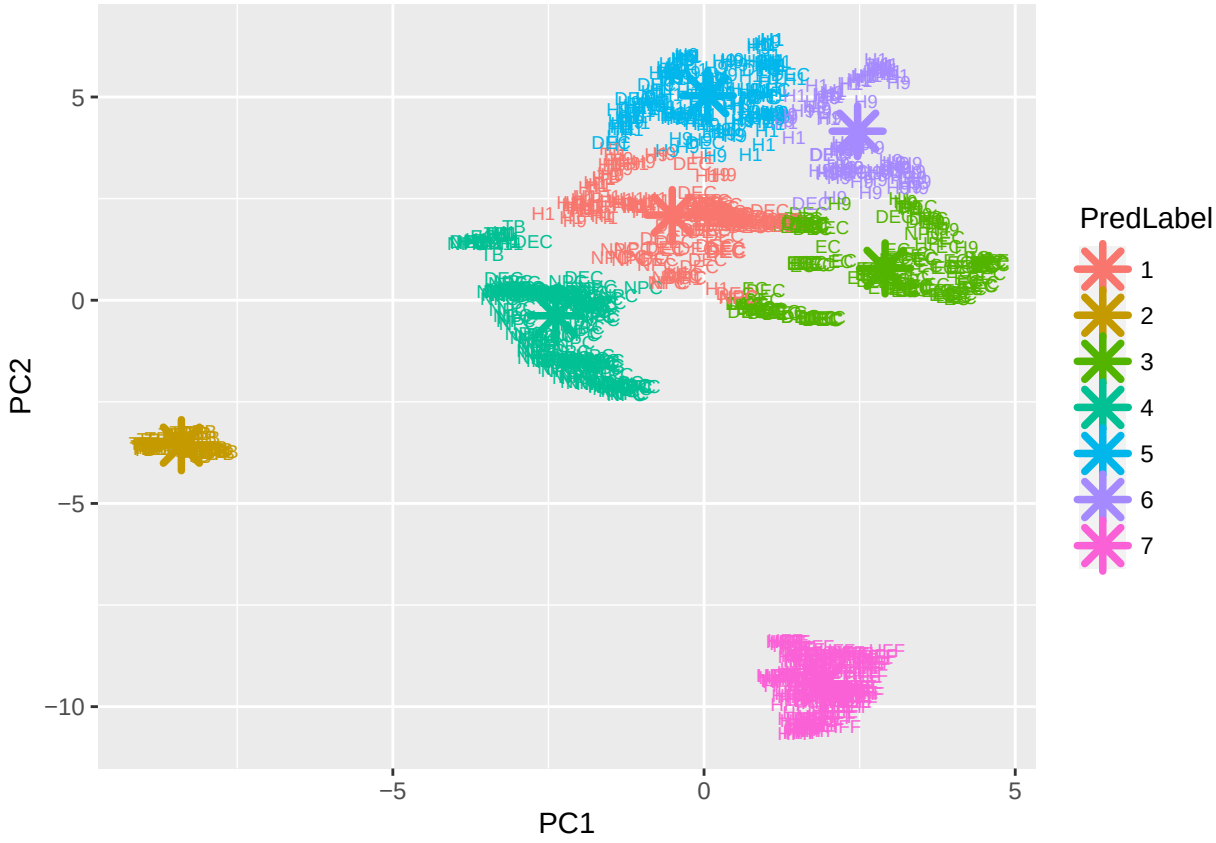
UMAP, then kmeans

```
library(umap)
```

```
chu.umap = umap(chu.x)
Z = chu.umap$layout

K = 7
km = kmeans(Z,centers=K)
```

```
clustered = data.frame(cbind(Z[,1],Z[,2],km$cluster,chu.y),stringsAsFactors = FALSE)
colnames(clustered) = c("PC1","PC2","PredLabel","CellTuning")
clustered$PC1 = as.numeric(clustered$PC1)
clustered$PC2 = as.numeric(clustered$PC2)
# projected k-means centers
group.data = data.frame(km$centers[,1:2])
```

```
group.data$label = rownames(group.data)
colnames(group.data) = c("PC1","PC2","PredLabel")

ggplot(clustered,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CellTuning), size = 2.5) +
  geom_point(data = group.data,size = 5, shape = 8, stroke = 2)
```




```
  theme(legend.position="none")
```

```
## List of 1
##  $ legend.position: chr "none"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```