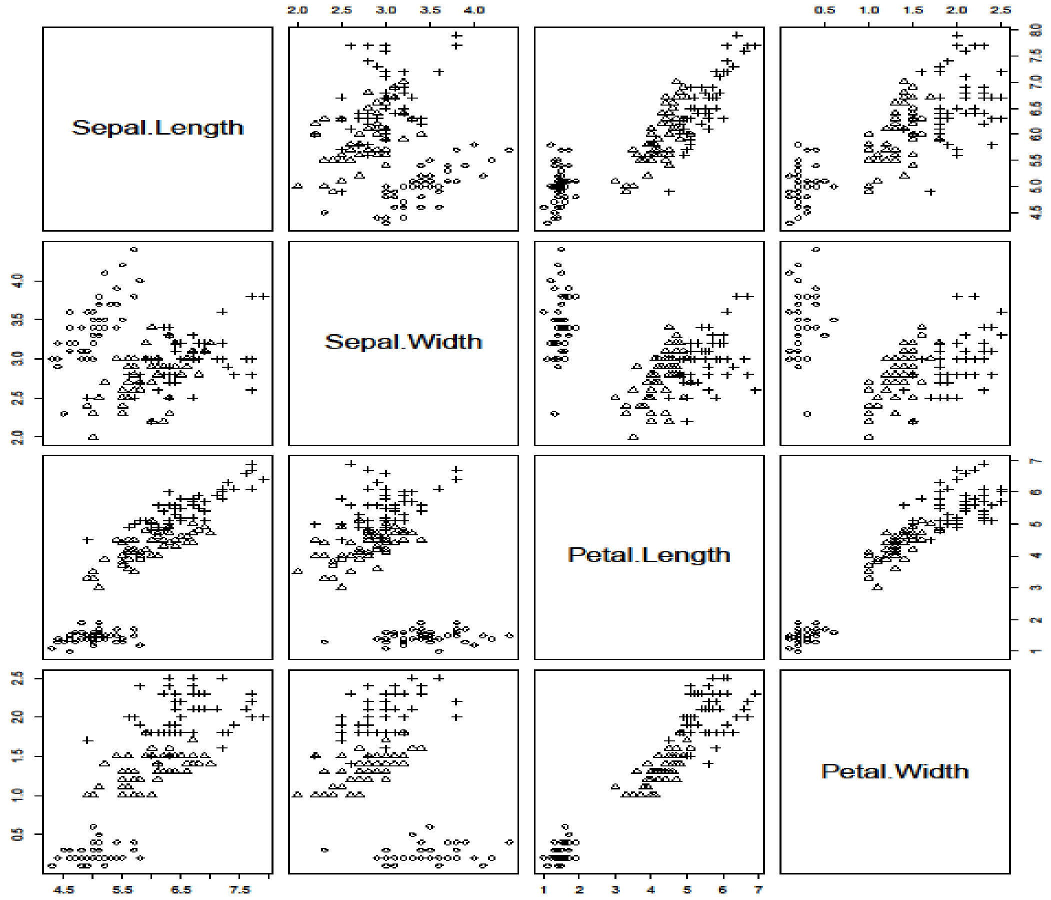
```
# EXAMPLE R CODE:   k-means in R    --- Fisher’s iris data      (by J. Corter)

# a simple example: clustering the iris data (N=150)
# we can read the iris data in from a text file:
#irisdat <- read.table("C:/Users/corter/Desktop/mdscstuff/IRIS_MLT.txt",header=T)
# OR, simply use the iris dataset that is pre-defined in R
iris
#NOTE: rownum=case number, cols 1:4 = data, col5 = class name

# plot the data points on all four variables:
groupnum<-rep(1:3,c(50,50,50))
plot(iris[,1:4],pch=as.numeric(groupnum))
```

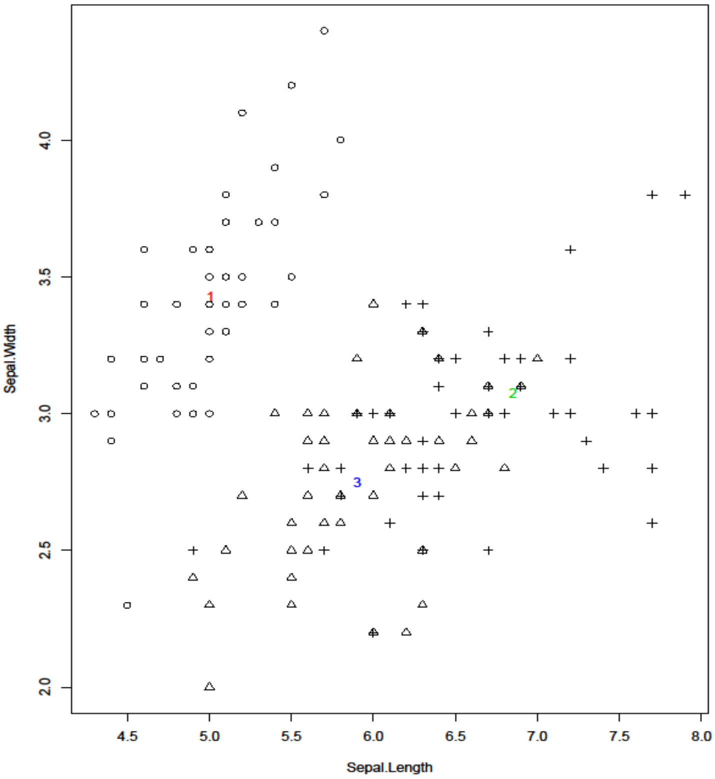


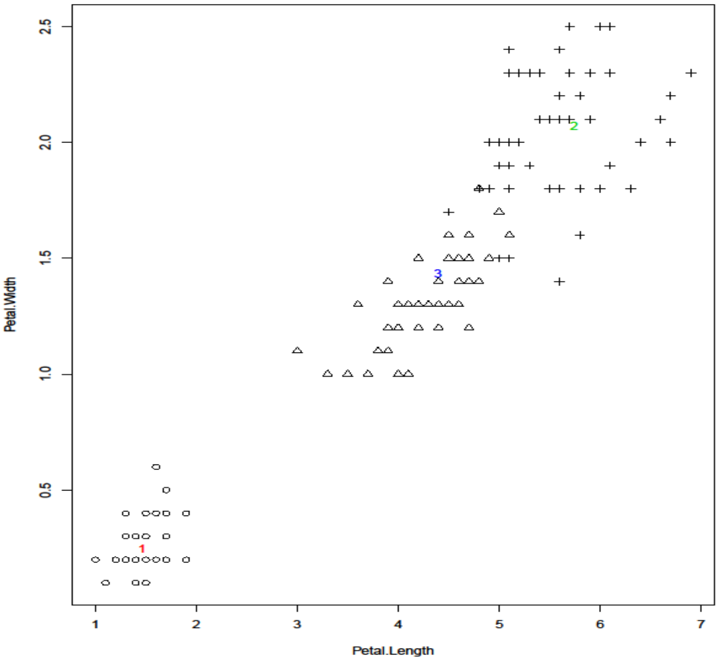
```
# run the k-means with k=3 clusters
# NOTE: the default in R's "kmeans" is to use 3 randomly selected cases as initial seeds
cl3 <- kmeans(iris[1:4],3)
```

```
>cl3
K-means clustering with 3 clusters of sizes 50, 38, 62

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     6.850000    3.073684     5.742105    2.071053
3     5.901613    2.748387     4.393548    1.433871
```

```
Clustering vector:
  [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 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 3 3 2 3 3
 [56] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2
[111] 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 2 2 2 2 3 2 2 2 3 2 2 2 3 2 2
3

Within cluster sum of squares by cluster:
[1] 15.15100 23.87947 39.82097
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betw
eenss"    "size"
[8] "iter"         "ifault"
> cl3$cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3
 [56] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2
[111] 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 2 2 2 2 3 2 2 2 3 2 2 2 3 2 2
3
> cl3$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     6.850000    3.073684     5.742105    2.071053
3     5.901613    2.748387     4.393548    1.433871
> cl3$totss
[1] 681.3706
> cl3$withinss
[1] 15.15100 23.87947 39.82097
> cl3$tot.withinss
[1] 78.85144
> cl3$betweenss
[1] 602.5192
> cl3$size
[1] 50 38 62
> cl3$iter
[1] 2
> cl3$ifault
[1] 0
```

```
# Now let's plot the data two dimensions at a time, and add cluster centroid points (plot the cluster numbers as text)
plot(iris[,1:2],pch=as.numeric(groupnum))
text(cl3$centers[1,1],cl3$centers[1,2],"1",col=10)
text(cl3$centers[2,1],cl3$centers[2,2],"2",col=11)
text(cl3$centers[3,1],cl3$centers[3,2],"3",col=12)
plot(iris[,3:4],pch=as.numeric(groupnum))
text(cl3$centers[1,3],cl3$centers[1,4],"1",col=10)
text(cl3$centers[2,3],cl3$centers[2,4],"2",col=11)
text(cl3$centers[3,3],cl3$centers[3,4],"3",col=12)
```


Sepal.Width
4.0
3.5
3.0
2.5
2.0
Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
1
2
3


Petal.Width
2.5
2.0
1.5
1.0
0.5
Petal.Length
1
2
3
4
5
6
7
1
2
3

```
#  option: we can request multiple random starts
#  but the manual is not quite clear as to what this accomplishes
# ideally, it would run 25 random starts, then save the BEST solution
cl <- kmeans(iris[,1:4], 3, nstart = 25)

# let's try 10 random starts (with k=3), and SAVE all the solutions to compare:
clmem <- rep(0,1500)
 dim(clmem) <- c(150,10)
clWSS <- rep(0,10)
for (i in 1:10)
{ cl <- kmeans(iris[,1:4],3)
  clmem[,i] <- cl$cluster
  clWSS[i] <- sum(cl$withinss)
}
# matrix “clmem” now holds the cluster solutions for these ten random starts:
clmem
# while vector "clWSS" hold the (WSS, summed across the 3 clusters) for the 10 random starts
clWSS

> clmem
       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
  [1,]    3    2    1    2    1    3    3    3    1     2
  [2,]    3    2    1    1    1    3    3    3    1     3
  [3,]    3    2    1    1    1    3    3    3    1     3
  [4,]    3    2    1    1    1    3    3    3    1     3
  [5,]    3    2    1    2    1    3    3    3    1     2
... (approx. 140 more rows here..)
[148,]    1    3    2    3    3    2    1    2    2     1
[149,]    1    3    2    3    3    2    1    2    2     1
[150,]    2    1    3    3    2    1    2    1    3     1
> # while vector "clWSS" hold the (WSS, summed across the 3 clusters) for the 10 random starts
> clWSS
 [1]  78.85144  78.85144  78.85144 142.75352  78.85144  78.85144  78.85144
 [8]  78.85144  78.85144 142.75352
# NOTE that two solutions stabilized at a local minimum (WSS=142.75352)
```