Self-Organizing Maps

Dataset: USArrests

Goal: Perform hierarchical clustering of the data using Euclidean distance
Fit the data into SOM and identify clusters in SOM
Compare results of hierarchical clustering with SOM

USArrests has four variables and different metrics

```
> data("USArrests")
> head(USArrests)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
> dim(USArrests)
[1] 50  4
```

Let us scale the data to make all variables are uniformly considered for clustering.

```
> USscaled<-scale(USArrests)
> dim(USscaled)
[1] 50  4
> head(USscaled)
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

Hierarchical clustering:

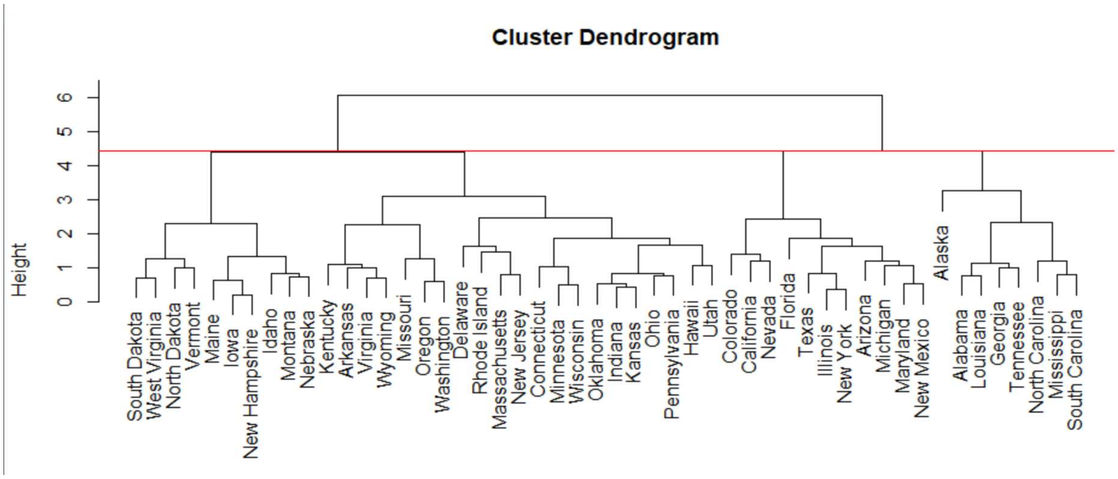
1. **Calculate a distance matrix for the scaled data using Euclidean distance**

```
> distmat<-dist(USscaled,method="euclidean")
> distmatx<-as.matrix(distmat)
> dim(distmatx)
[1] 50 50
```

It calculated distance for each state to each other state.

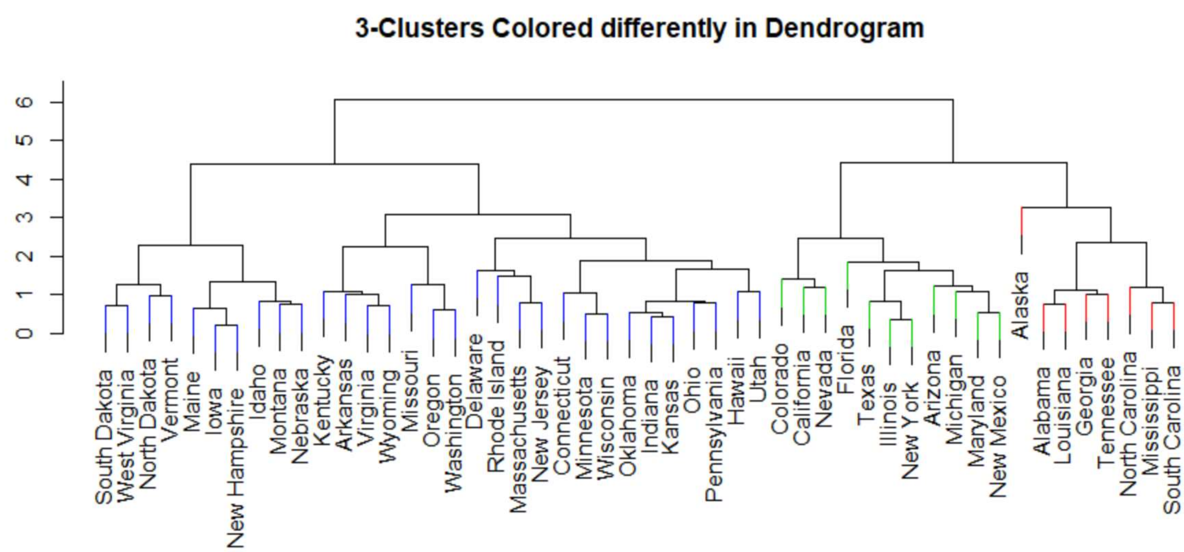
2. **Cluster using method as complete**

```
> hc<-hclust(distmat,method="complete")
> plot(hc)
> abline(h=4.42,col="red")
```



3. **Cut the tree at height 4.42 gives three clusters and similar to cut with k=3**

```
> hcut<-cutree(hc,h=4.42)
> table(hcut)
hcut
 1  2  3
 8 11 31
>
> kcut<-cutree(hc,k=3)
> table(kcut)
kcut
 1  2  3
 8 11 31
```



Self-Organizing maps: Fit the same data using SOM and cluster into 3 groups

```
> dim(USscaled)
[1] 50  4
> head(USscaled)
               Murder   Assault   UrbanPop         Rape
```

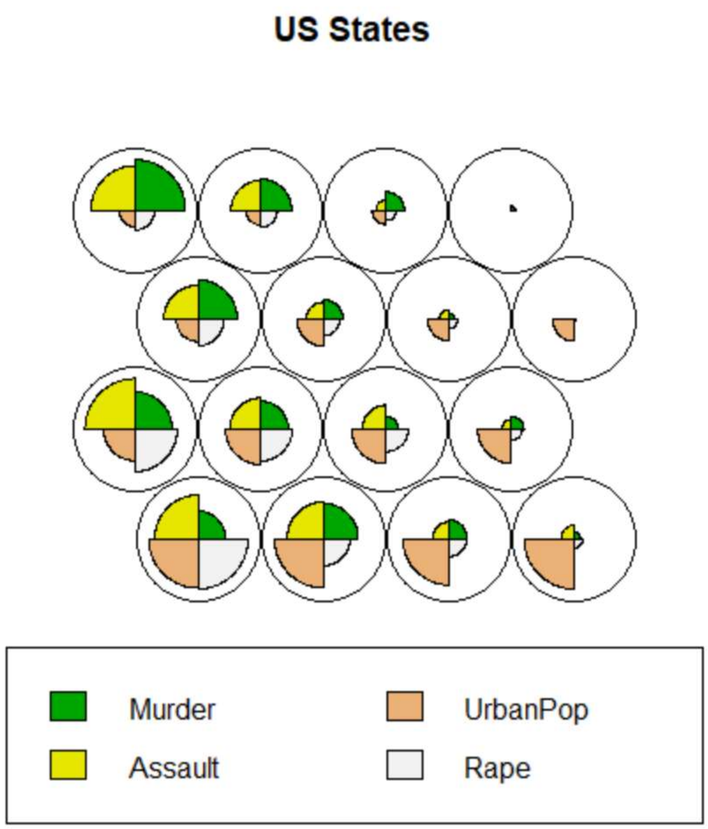
```
Alabama     1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska      0.50786248 1.1068225 -1.2117642  2.484202941
Arizona     0.07163341 1.4788032  0.9989801  1.042878388
Arkansas    0.23234938 0.2308680 -1.0735927 -0.184916602
California  0.27826823 1.2628144  1.7589234  2.067820292
Colorado    0.02571456 0.3988593  0.8608085  1.864967207
```

1. **Prepare a grid of 4*4 as total records is 50 so 16 prototypes is enough to fit the whole data**

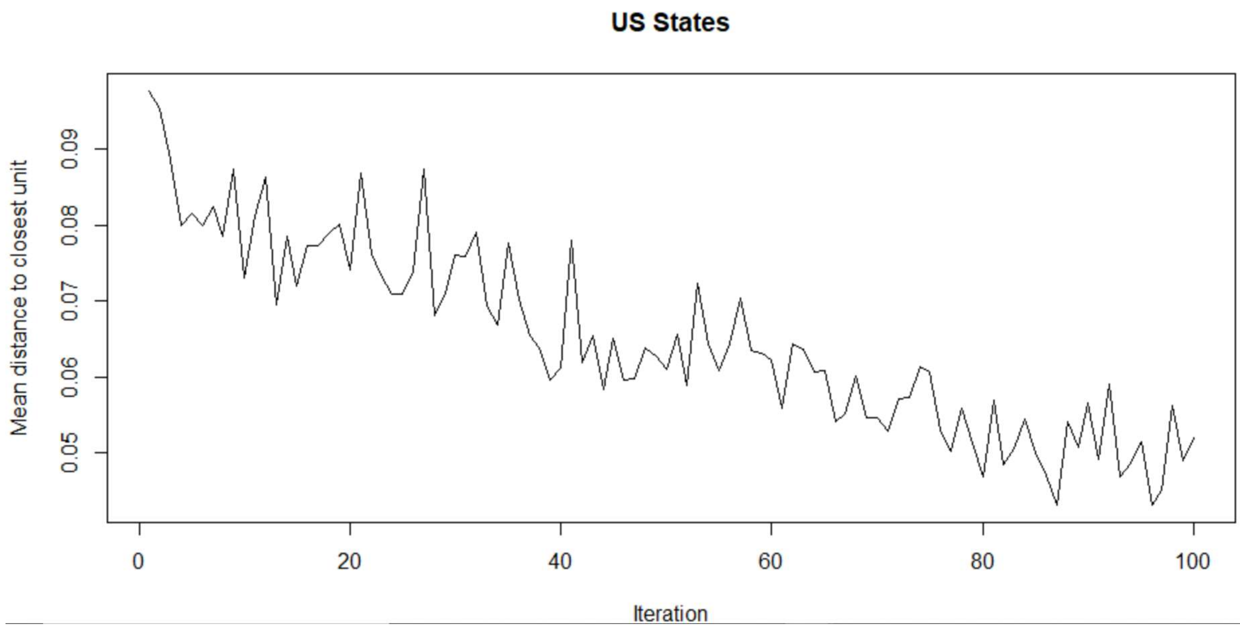
```
> us.sgrid<-somgrid(xdim=4,ydim=4,topo=c("hexagonal"))
```

2. **Fit data into SOM and with iterations for training data – 100 as there are 16 prototypes even though number of records is 50 so you should run few iterations to set the prototypes- build code vectors correctly**

```
> us.som<-som(USscaled,us.sgrid,rlen=100)
> codes <- us.som$codes[[1]]
> plot(us.som, main = "US States")
```

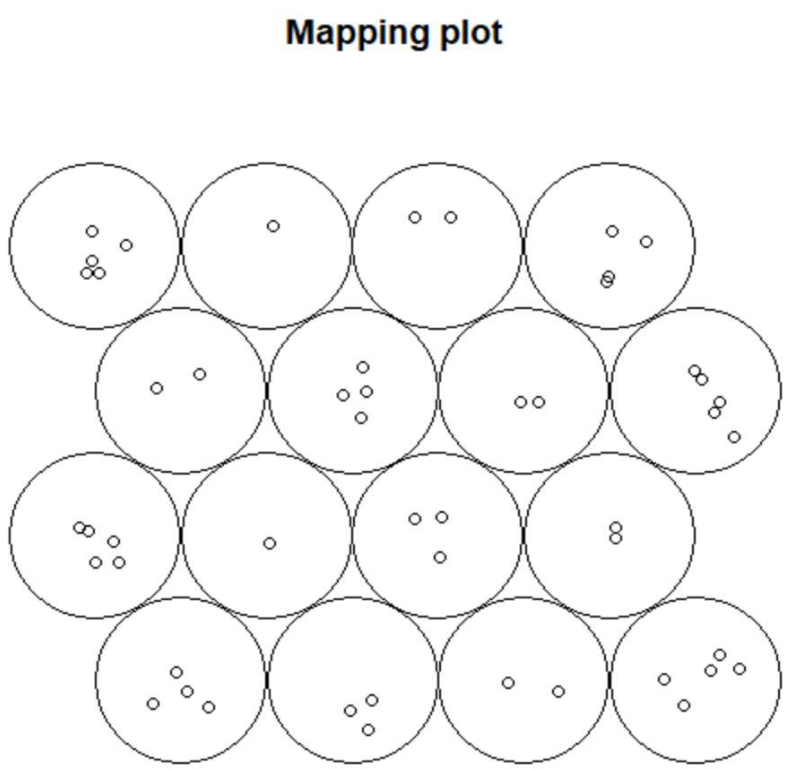


Changes plot:

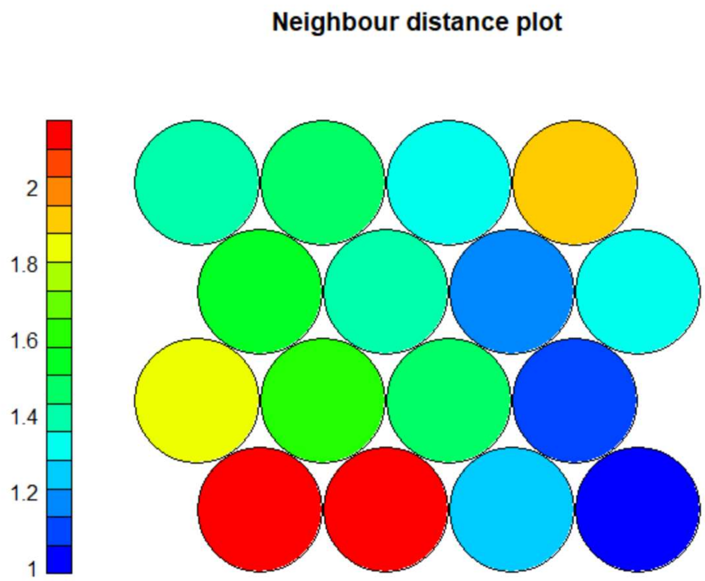


From this we can infer that 86-90 iterations should have been enough after that again there was spike and code vectors were moving away from each other. So, Let’s see how it behaves at the end and then we can change parameter.

Mapping plot: grids are well defined each grid has at least one state in it.

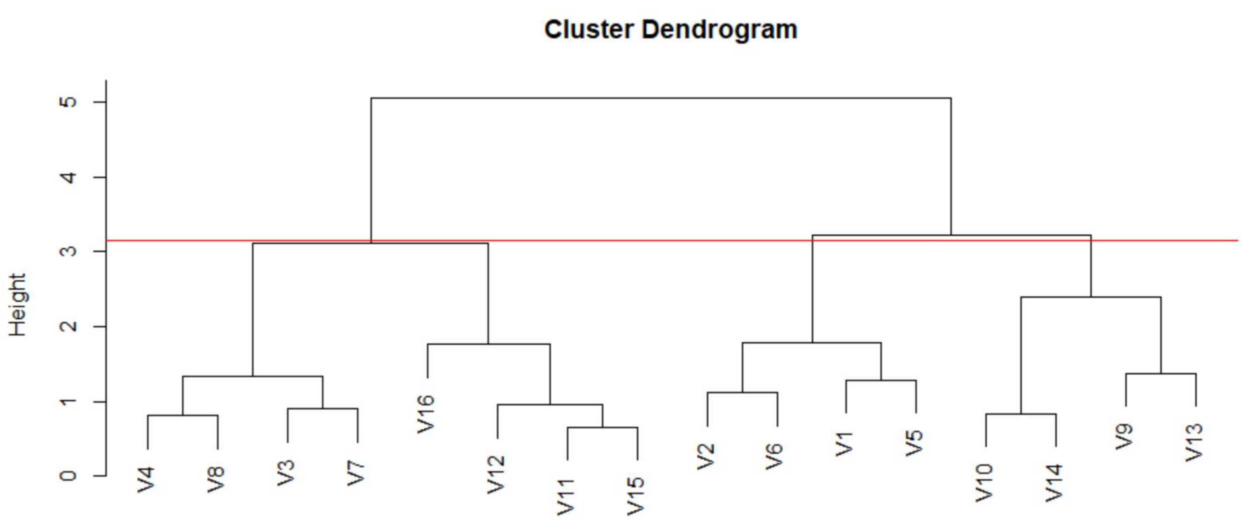


Neighbor distance plot: Most grids are near to each other.



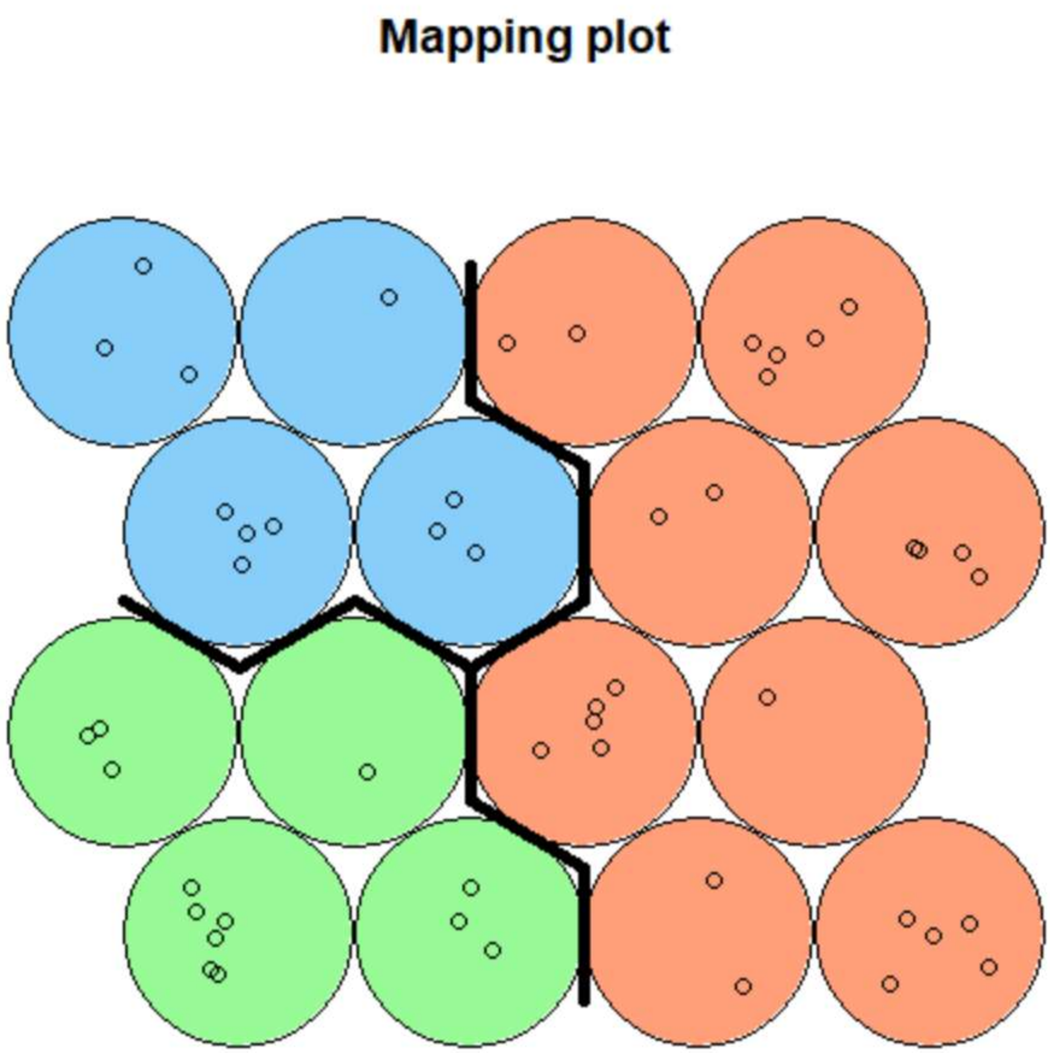
Cluster the grids by using the code vectors:

```
> d <- dist(codes)
> hc <- hclust(d)
>
> plot(hc)
```



Code vectors are clustered clearly. Let's cut to at 3.15 to create 3 clusters and then compare it with hierarchical clustering.

```
> som_cluster <- cutree(hc, h = 3.15)
>
> # plot the SOM with the found clusters
>
> my_pal <- c("palegreen", "lightsalmon", "lightskyblue","orange")
> my_bhcol <- my_pal[som_cluster]
>
> plot(us.som, type = "mapping", col = "black", bgcol = my_bhcol)
> add.cluster.boundaries(us.som, som_cluster)
> table(som_cluster)
som_cluster
1 2 3
4 8 4
```



Comparison of SOM fit clustering with regular Hierarchical clustering:

SOM fit clustering	
State	**Cluster**
Alaska	1
Arizona	1
California	1
Colorado	1
Florida	1
Illinois	1
Maryland	1
Michigan	1
Missouri	1
Nevada	1
New Mexico	1
New York	1
Texas	1
Connecticut	2
Delaware	2
Hawaii	2
Idaho	2
Indiana	2
Iowa	2
Kansas	2
Maine	2
Massachusetts	2
Minnesota	2
Montana	2
Nebraska	2
New Hampshire	2
New Jersey	2
North Dakota	2
Ohio	2
Oklahoma	2
Oregon	2
Pennsylvania	2
Rhode Island	2
South Dakota	2
Utah	2
Vermont	2
Washington	2
West Virginia	2
Wisconsin	2
Alabama	3
Arkansas	3
Georgia	3
Kentucky	3
Louisiana	3
Mississippi	3
North Carolina	3
South Carolina	3
Tennessee	3
Virginia	3
Wyoming	3

Cluster1	**13**
Cluster2	**26**
Cluster3	**11**

Hierarchical Clustering - Complete	
State	**Cluster**
Alabama	1
Alaska	1
Georgia	1
Louisiana	1
Mississippi	1
North Carolina	1
South Carolina	1
Tennessee	1
Arizona	2
California	2
Colorado	2
Florida	2
Illinois	2
Maryland	2
Michigan	2
Nevada	2
New Mexico	2
New York	2
Texas	2
Arkansas	3
Connecticut	3
Delaware	3
Hawaii	3
Idaho	3
Indiana	3
Iowa	3
Kansas	3
Kentucky	3
Maine	3
Massachusetts	3
Minnesota	3
Missouri	3
Montana	3
Nebraska	3
New Hampshire	3
New Jersey	3
North Dakota	3
Ohio	3
Oklahoma	3
Oregon	3
Pennsylvania	3
Rhode Island	3
South Dakota	3
Utah	3
Vermont	3
Virginia	3
Washington	3
West Virginia	3
Wisconsin	3
Wyoming	3

Cluster1	**8**
Cluster2	**11**
Cluster3	**31**

Observation:

When you re-run the data of hierarchical clustering will remain same but the SOM clustering will change as the training data will be picked random again and the code vectors are reformed as data is really small and few variables so change in code vectors distance vectors will differ to greater extent so there by results in different clustering.

	Advantages of Hierarchical clustering over SOM
1	Hierarchical clustering depends on the distance matrix of all variables scaled by which actual distance between objects can be defined and clustering gives actual results where as in SOM it purely depends on the code vectors and the learning rate (training data) used to derive code vectors which if not appropriate may lead to wrong SOM fit
2	It's easy to cut a tree by height or by k for further analysis and visually perfect using dendrogram but it's not that great as it does not include values of the clusters as its possible to view the range of values in SOM
3	If goal is clustering than its good to use Hierarchical clustering over SOM

High dimensional data viewing in 2D to understand the grouping and visualization and use it for prediction so SOM offers so much more than hierarchical clustering and SOM behave badly for smaller datasets as learning rate and to build code vectors will not be perfect. It is much suitable for high dimensional data.