Table of Contents

Section 4. Unsupervised Learning

4.1 Learning outcomes

The following topics were studied and practiced using some practice exercises and applied to the current project

- The challenge of Unsupervised learning
- Principal component analysis
    - Principal components
    - Feature extraction and representation
- Clustering methods
    - K-Means clustering
    - Hierarchical clustering

4.2 Project 4: Principal Components Analysis and clustering of USArrests data

Objective:

The object of this project is to perform Principal component analysis on a small data set and to be able to use k-means clustering to divide the data into clusters.

Our main goal is to:
1. Identify the variations in data and reduce the dimensionality. Identify new underlying meaningful variables or components
2. Explain graphically the top components that are responsible for maximum proportion of variance explained (PVE)
3. Group the 50 states into clusters based on the main principle components

Data used:

" USArrests"
For this project, we will analyze USArrests data(50 states and 4 crime categories) that is available in R. Each row here is a state and each column is a crime category.

A data frame with 50 observations on 4 variables.

[1]	Murder	numeric	Murder arrests (per 100,000)
[2]	Assault	numeric	Assault arrests (per 100,000)
[3]	UrbanPop	numeric	Percent urban population
[4]	Rape	numeric	Rape arrests (per 100,000)

Tools used:

R Studio

Language:

R

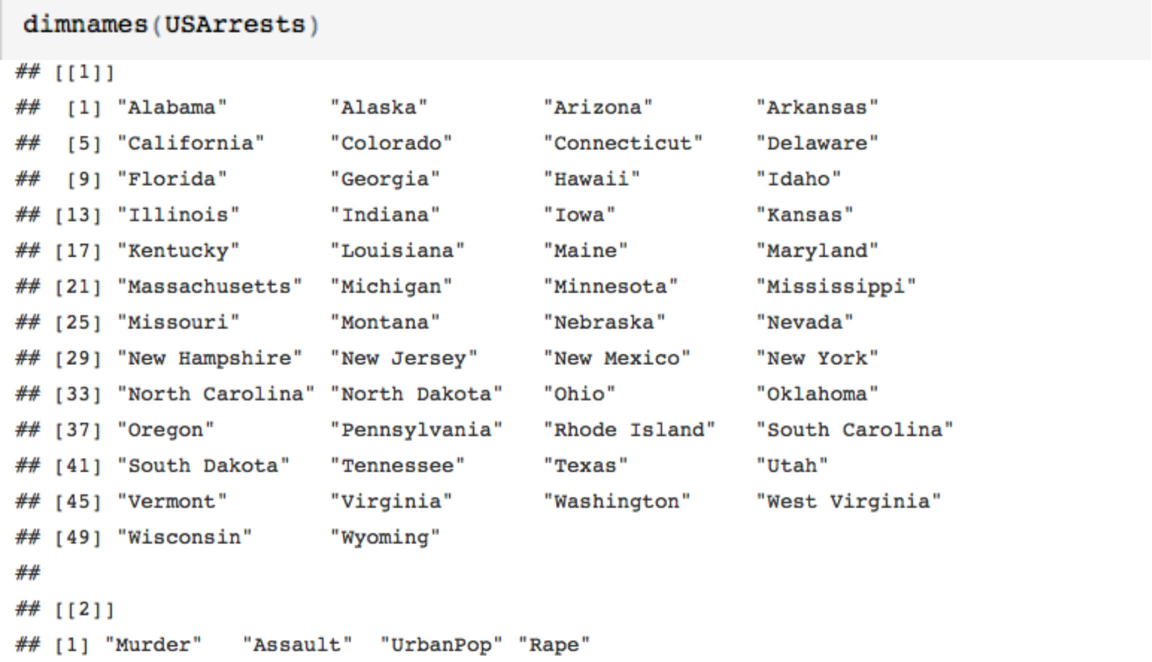
Analysis:

Part 1. We will first apply kmeans clustering to our original data by varying k and finding out the ideal k using elbow graph. Then use the new K to group the states into natural clusters.

Part 2. We will then apply PCA, understand the underlying new components and use the new reduced dimensional data to perform kmeans clustering. Here also, we will use the ideal k suggested by the elbow graph. We will finally cluster the states using the new K and analyze the results or try to understand the groups formed.

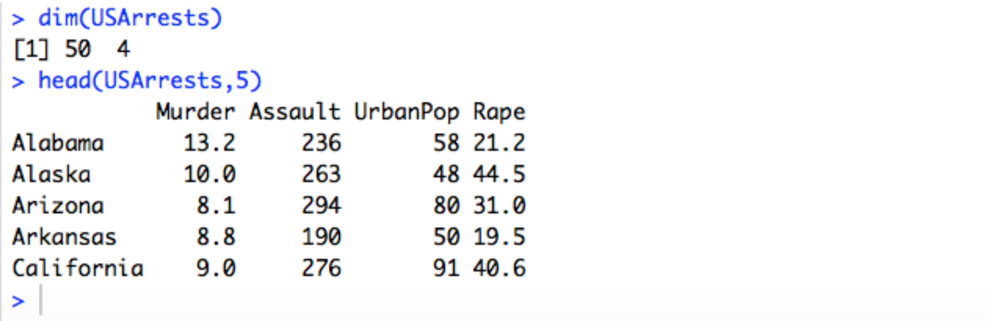
4.2.1 Part 1: K-Means Clustering on Original USArrests data

STEP 1: LOOKING AT THE DATA



```
dimnames(USArrests)
```

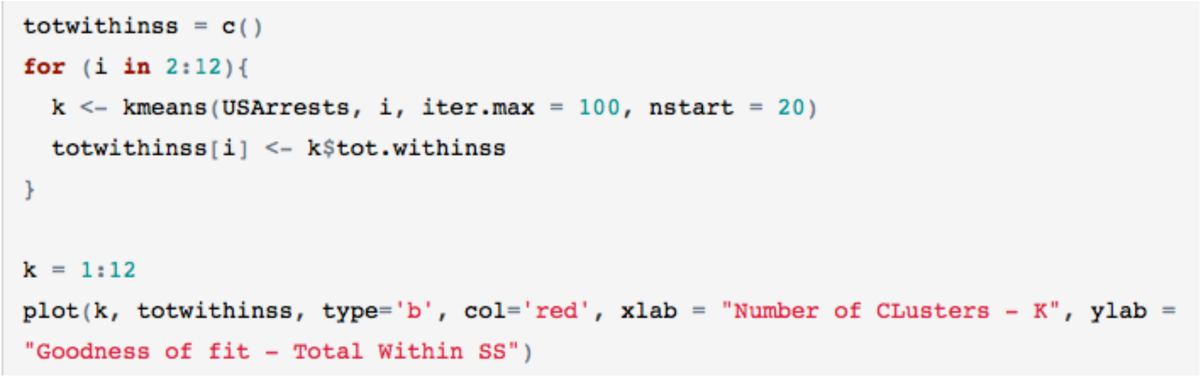
```
## [[1]]
##  [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
##  [5] "California"     "Colorado"       "Connecticut"    "Delaware"
##  [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
## [13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
## [17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
## [21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
## [25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
## [29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
## [33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
## [37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
## [41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
## [45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
## [49] "Wisconsin"      "Wyoming"
##
## [[2]]
## [1] "Murder"   "Assault"  "UrbanPop" "Rape"
```



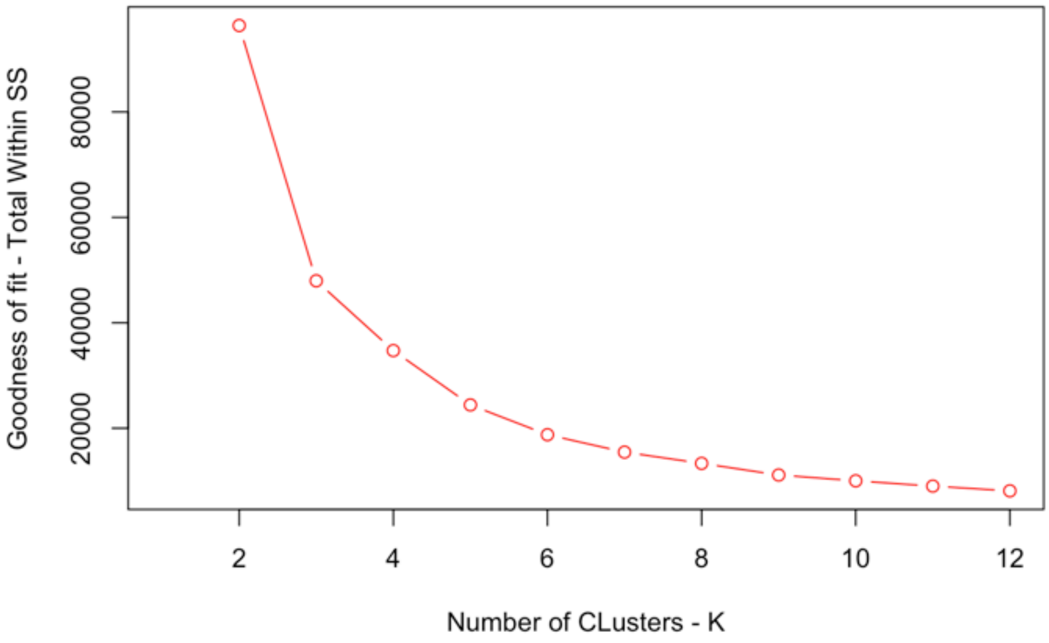
```
> dim(USArrests)
[1] 50  4
> head(USArrests,5)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
> 
```

STEP 2: PERFORMING KMEANS AS A PRE-PROCESSING AND FINDING K FROM ELBOW GRAPH OF WITHINSS

In order to find the ideal number of clusters, we have to look at the parameter withinss returned by the kmeans object as a preprocessing step. The lesser the "withinss" the better the clustering. So first, lets do kmeans for k = 2 to 15 iteratively and note down the ***Total within sum of squares*** (tot.withiss) for each kmeans object. Then plot the "total-within-ss vs. K" and find the right "K" at the elbow of the scree plot.



```
totwithinss = c()
for (i in 2:12){
  k <- kmeans(USArrests, i, iter.max = 100, nstart = 20)
  totwithinss[i] <- k$tot.withinss
}

k = 1:12
plot(k, totwithinss, type='b', col='red', xlab = "Number of CLusters - K", ylab =
"Goodness of fit - Total Within SS")
```

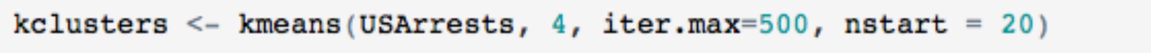


The value of K at the elbow ***suggests K=4*** might be the natural number of clusters that can represent our data without too many overlaps.

STEP 3: FINAL CLUSTERING OF THE STATES AND PLOTTING THE RESULTS

Using kmeans() again to form a kmeans object with the K value 4. That is, we are forming 4 clusters of the 50 states in the data.

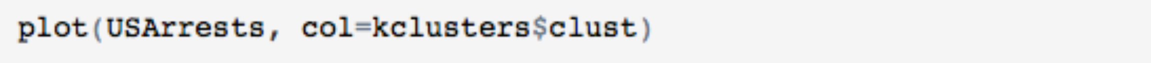
```
#clustering the original data with kmeans using the ideal k
value (k=4)
```



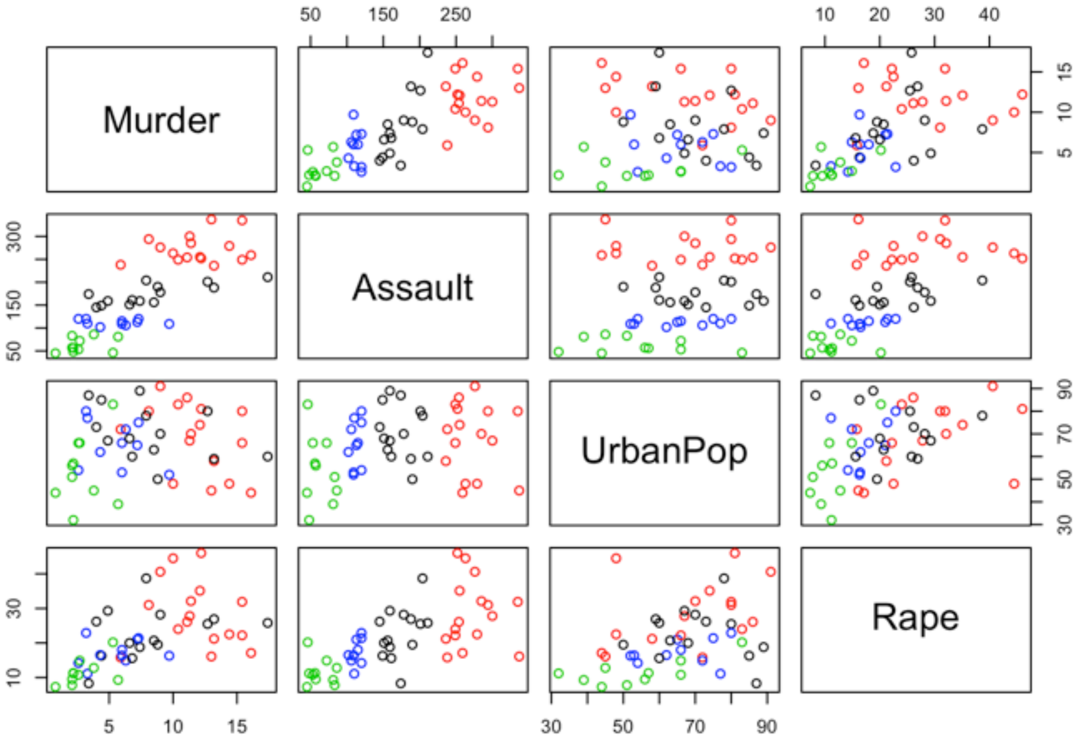
```
kclusters <- kmeans(USArrests, 4, iter.max=500, nstart = 20)
```

The kclusters object created above will contain a vector called "clusters" that will tell us which points or precisely states belong to which clusters.

So lets plot the original data with all 4 variables (Murder, Assault, UrbanPop and Rape) and color them as per the clusters they form:



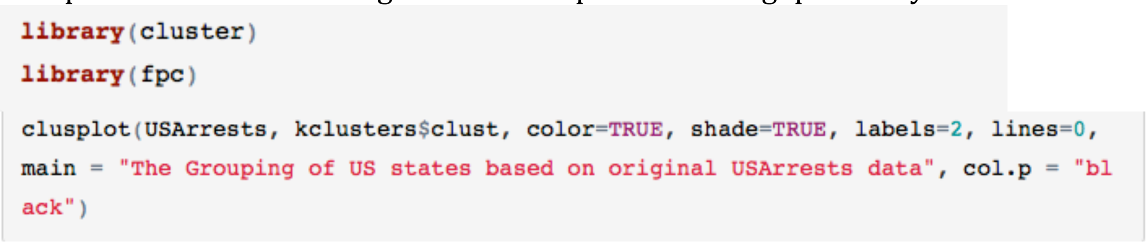
```
plot(USArrests, col=kclusters$clust)
```



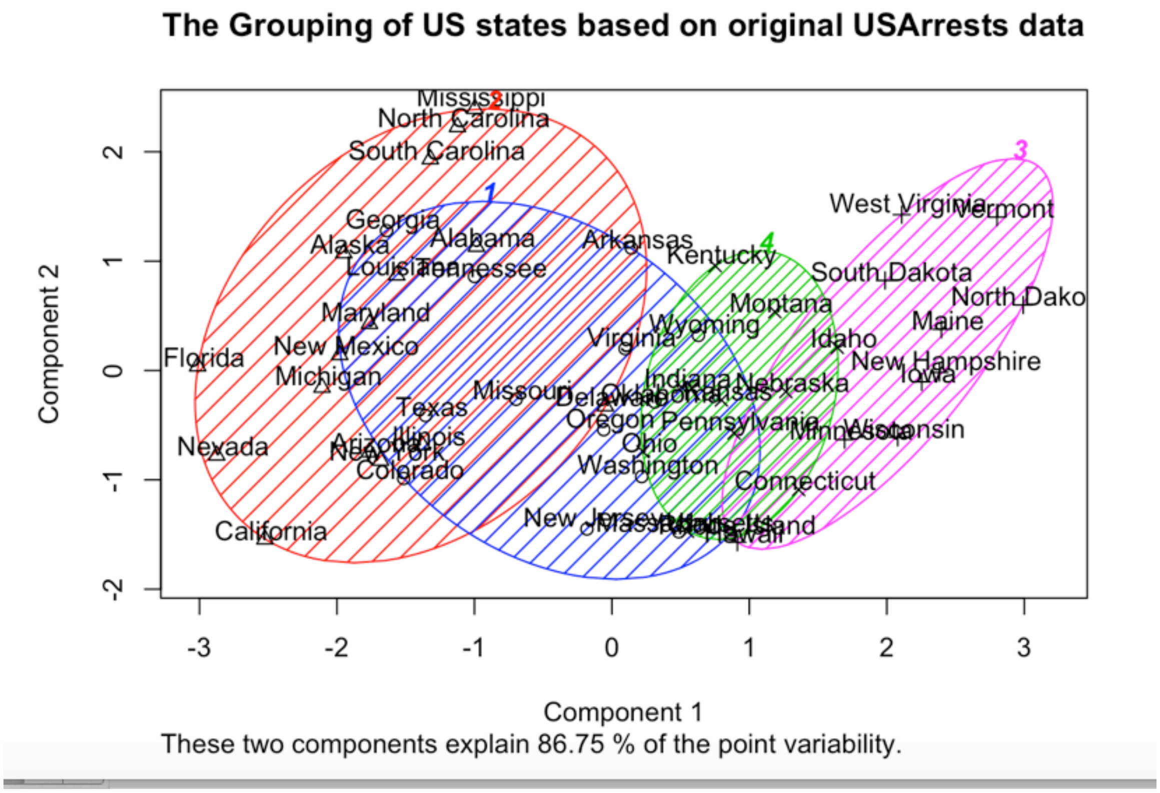
We can see that there are 4 clusters formed for each of the pairs of variables.
We can say:
(i) Murder vs Assault is very clear and neat as the states have formed distinct clusters telling us which states are low on assault and murder and which states are very high.

(ii) But, if we see the relation between UrbanPop and any other variable or Rape and any other variable, our clusters are not distinct enough.
We cannot conclusively derive any meaning from these clusters yet.

Lets plot all the variables together in a 2D plot now using fpc library:



```
library(cluster)
library(fpc)
clusplot(USArrests, kclusters$clust, color=TRUE, shade=TRUE, labels=2, lines=0,
main = "The Grouping of US states based on original USArrests data", col.p = "bl
ack")
```



Can we interpret what the clusters are trying to say?

Because this is unsupervised learning, we do not have an actual recorded response label that can tell us what those groups mean. But clustering will group naturally closer points together suggestively. We should be able to interpret on the scale as to what groups mean.

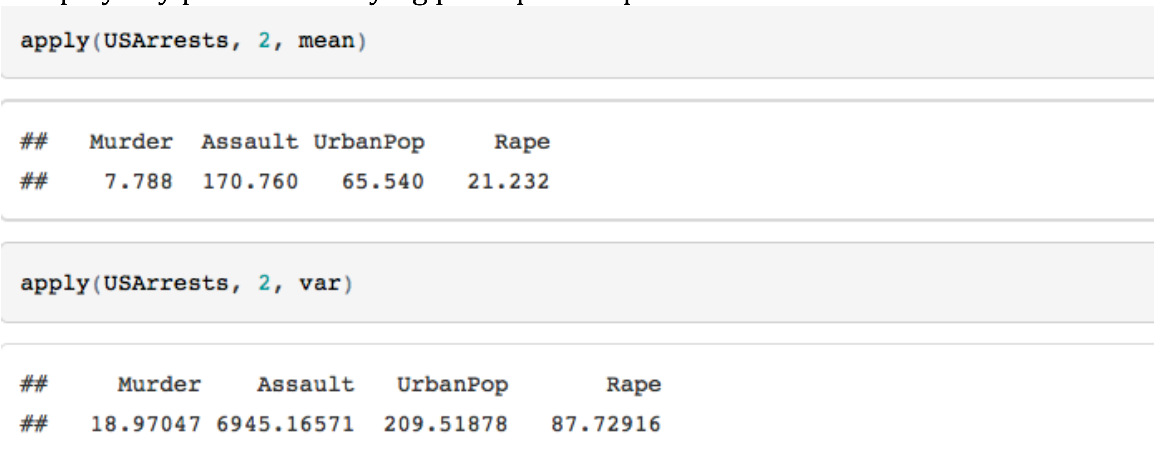
Here the states over lap a lot on the scale of the 4 variables suggesting everything everywhere

The next part is an attempt to simplify the interpretation process by identifying some underlying meaning variables from the original variables and use them instead to perform k-means clustering. For this we will use Principal component Analysis and reduce the dimensions. This is especially useful when we have gnome expression data where there are 100s of observations but millions of variables. PCA in this case, will identify 10 utmost variables that explain the maximum variances in the data and use those new components to cluster the data.

4.2.2 Part 2: K-Means Clustering using Principal Components

STEP 1: PERFORMING PCA AND IDENTIFYING THE PRINCIPAL COMPONENTS WITH HIGHEST PVE

Lets look at the means and variances within each variable (i.e. column – Murder, Assault, UrbanPop, Rape) to see how diverse the data is and to understand if scaling will play any part in identifying principal components.



```
apply(USArrests, 2, mean)
```

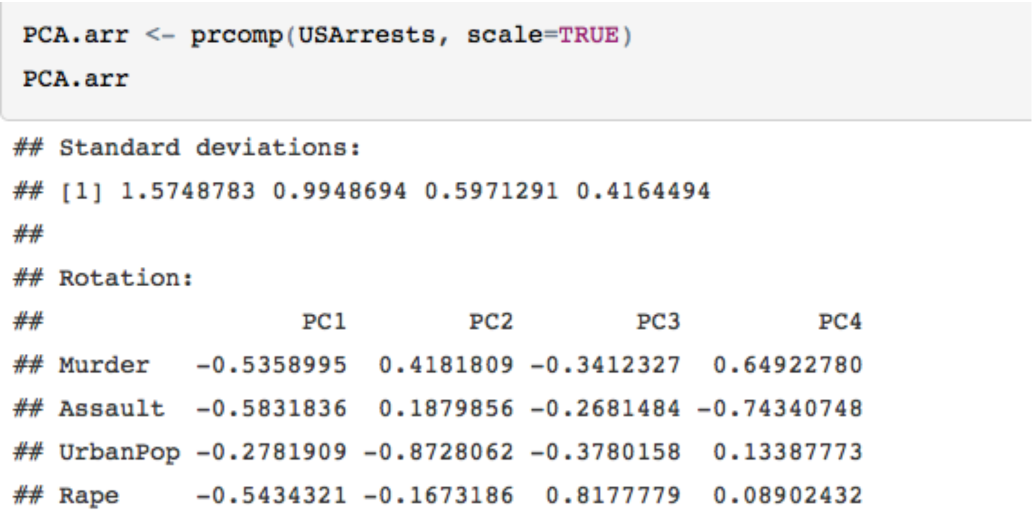
```
##   Murder  Assault UrbanPop     Rape
##    7.788  170.760   65.540   21.232
```

```
apply(USArrests, 2, var)
```

```
##     Murder    Assault   UrbanPop       Rape
##   18.97047 6945.16571  209.51878   87.72916
```

Assault has a very high variance and will pretty much become the only principal component that contributes to the proportion of variation explained (PVE). We do not want that. We need components that explained the maximum variation across all the data.

These huge differences in variances can be due to different units of the variables, so we will standardize the variables. We can do that by passing scale = True in the prcomp() method, so that every column has variance = 1 and mean = 0.

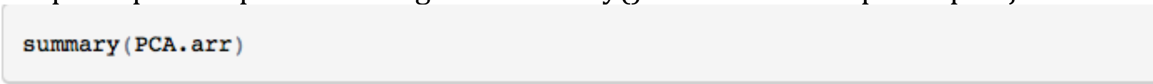


```r
PCA.arr <- prcomp(USArrests, scale=TRUE)
PCA.arr
```

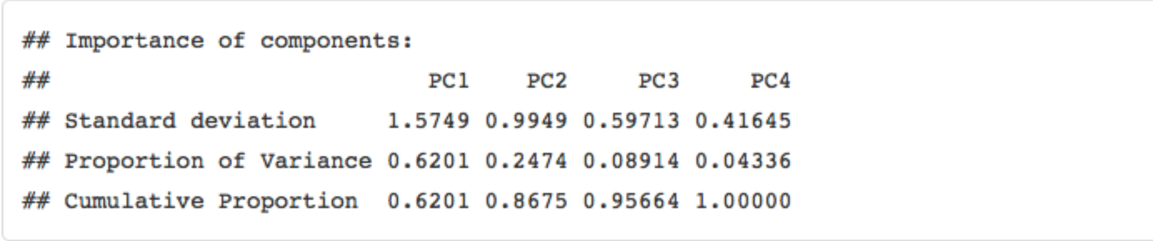
```
## Standard deviations:
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
##
## Rotation:
##                 PC1        PC2        PC3         PC4
## Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
## Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
## UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
## Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

Now we have PC1, PC2, PC3 and PC4 as new variables instead of our old - Murder, Assault, UrbanPop and Rape. These new variables are called Principal components. In the summary above, the **Standard deviation** values when squared will give us two of the components that explain most of the variance in the data. And then the **Rotation** represent loadings. For example PC1 is loaded on three of the four crimes and less loaded on the fourth one (based on the values). Also we notice that second principal component PC2 is heavily loaded on a single variable 'UrbanPop'

Also, We can obtain a summary of the proportion of variance explained (PVE) of all the principal components using the summary() method for the prcomp object

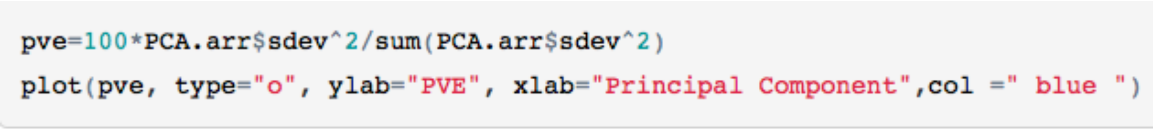


```r
summary(PCA.arr)
```

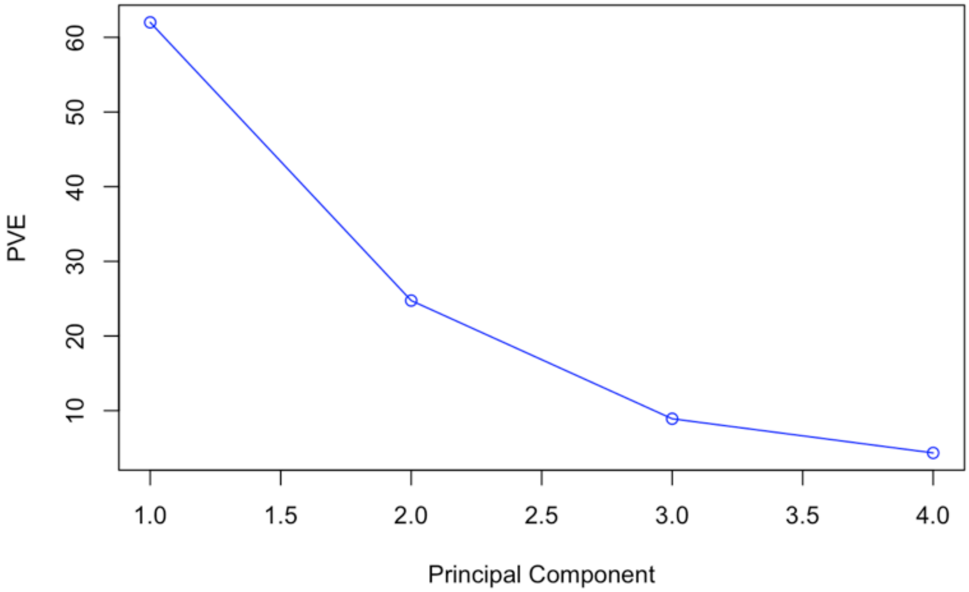


```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5749 0.9949 0.59713 0.41645
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336
## Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

It is realy informative to plot the PVE of each principal component (i.e. a scree plot). This along with the importance of components from above, will indicate how many principal components we will need.



```r
pve=100*PCA.arr$sdev^2/sum(PCA.arr$sdev^2)
plot(pve, type="o", ylab="PVE", xlab="Principal Component",col =" blue ")
```

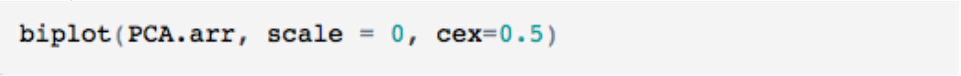


The elbow of the scree plot will indicate the ideal number of Principal components that will suffice for our representation. Here, the elbow indicates 2.

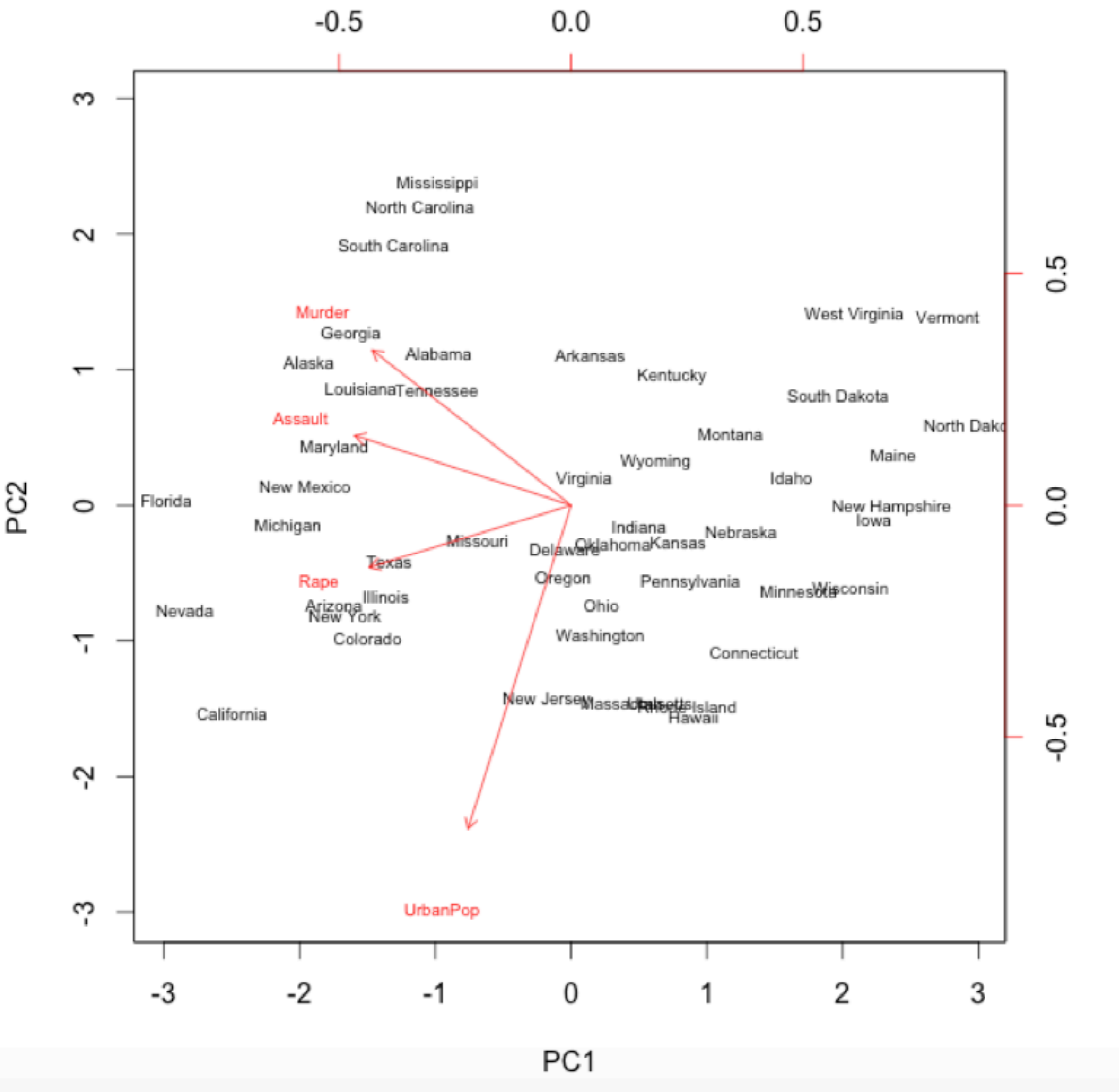
From the Importance of components table above, Proportion of variance explained by PC1 and PC2 together is 62% + 25%, that is 87%. We usually pick the components that explain >85% variance. ***So we need only the first two principal components PC1 and PC2, which pretty much sum up the scenario: - PC1 represents the total crime and PC2 represents the Urban Population***

Step 2: Analyzing the BIPLOT

We can also plot the Principal components using a biplot – which will represent very neatly, the two axes that best describe the variances in the data



```
biplot(PCA.arr, scale = 0, cex=0.5)
```



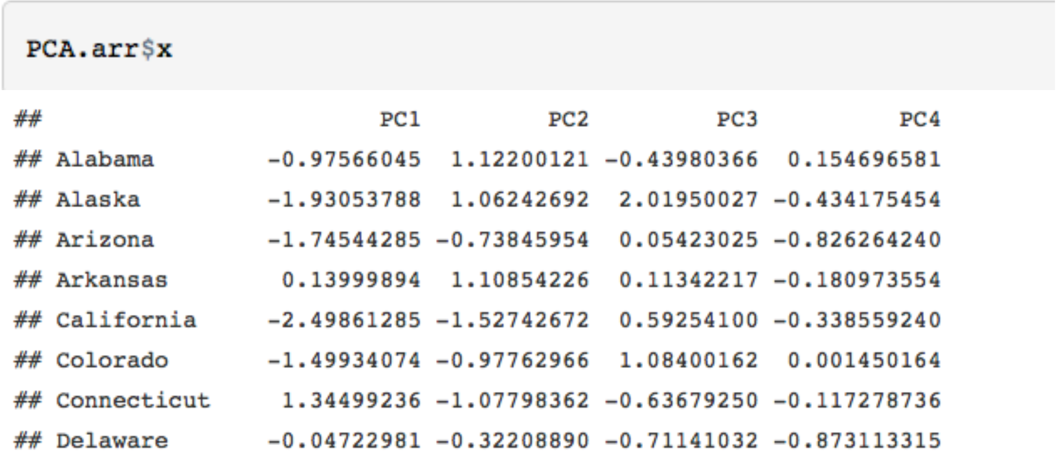
1. The Vectors in Red represent the “Directions of the loadings” for the principal components.
2. The first axis is largely due to three crimes - murder, assault and rape.
3. The second axis is solely due to UrbanPop

So the states on the side of the direction vectors show indications of high crime - for eg: Nevada, California, Michigan, New Mexico etc have high total crime while on the other side, West Virgiia, North Dakota, Marine etc have low total crime

Similarly, for UrbanPop, we can say New Jersy, Rhode ISland, Hawaii have a high Urban Population with Arkansas, Mississipi, North Carolina on the other side of UrbanPop scale meaning they are rural.

STEP 3: PLOTTING ELBOW GRAPH OF WITHINSS AND IDENTIFYING THE IDEAL K

We can access the new data points of the principal components using the prcomp object's 'x' parameter as follows:



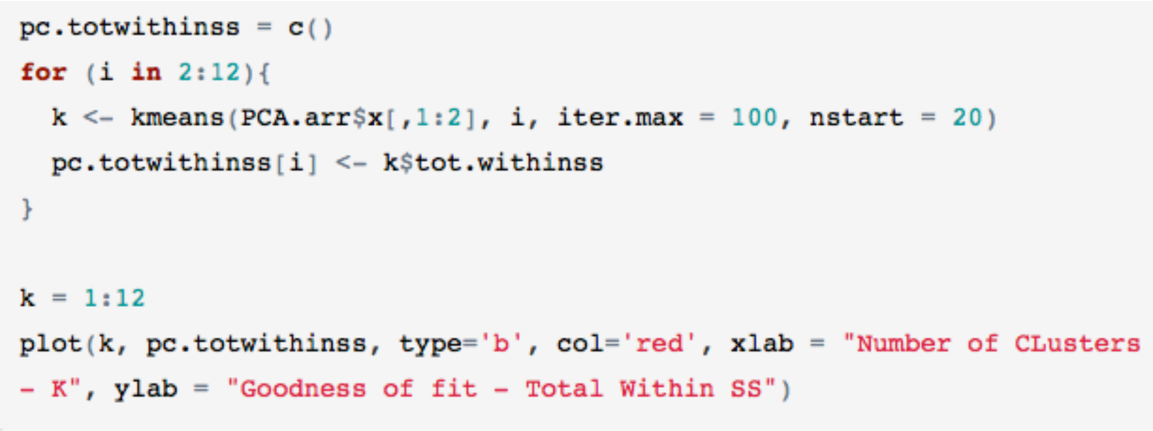
```
PCA.arr$x
```

```
##                        PC1         PC2         PC3          PC4
## Alabama        -0.97566045  1.12200121 -0.43980366  0.154696581
## Alaska         -1.93053788  1.06242692  2.01950027 -0.434175454
## Arizona        -1.74544285 -0.73845954  0.05423025 -0.826264240
## Arkansas        0.13999894  1.10854226  0.11342217 -0.180973554
## California     -2.49861285 -1.52742672  0.59254100 -0.338559240
## Colorado       -1.49934074 -0.97762966  1.08400162  0.001450164
## Connecticut     1.34499236 -1.07798362 -0.63679250 -0.117278736
## Delaware       -0.04722981 -0.32208890 -0.71141032 -0.873113315
```

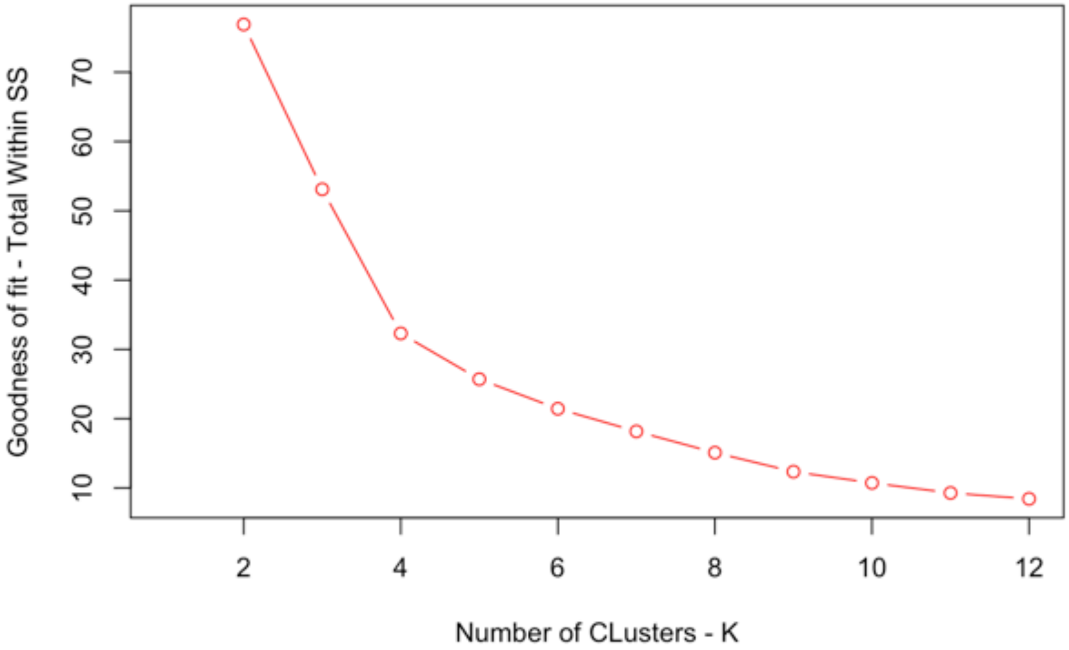
….

…

Now, lets repeat the same pre-processing we did earlier using kmeans() to identify the ideal k from the elbow graph, but this time, using the new principal components PC1 and PC2. Then plot the elbow graph



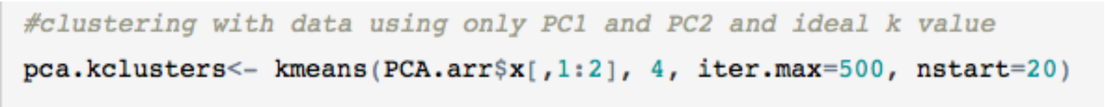
```
pc.totwithinss = c()
for (i in 2:12){
  k <- kmeans(PCA.arr$x[,1:2], i, iter.max = 100, nstart = 20)
  pc.totwithinss[i] <- k$tot.withinss
}

k = 1:12
plot(k, pc.totwithinss, type='b', col='red', xlab = "Number of CLusters
- K", ylab = "Goodness of fit - Total Within SS")
```



Here the number of natural clusters suggested at the elbow is also ***K=4***

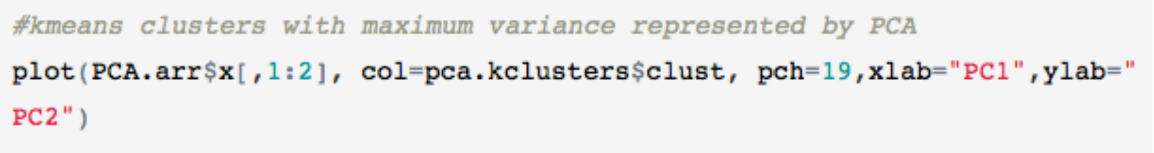
Step 4: Final clustering of the states and plotting the results

Finally lets repeat the formation of clusters using the new PCs

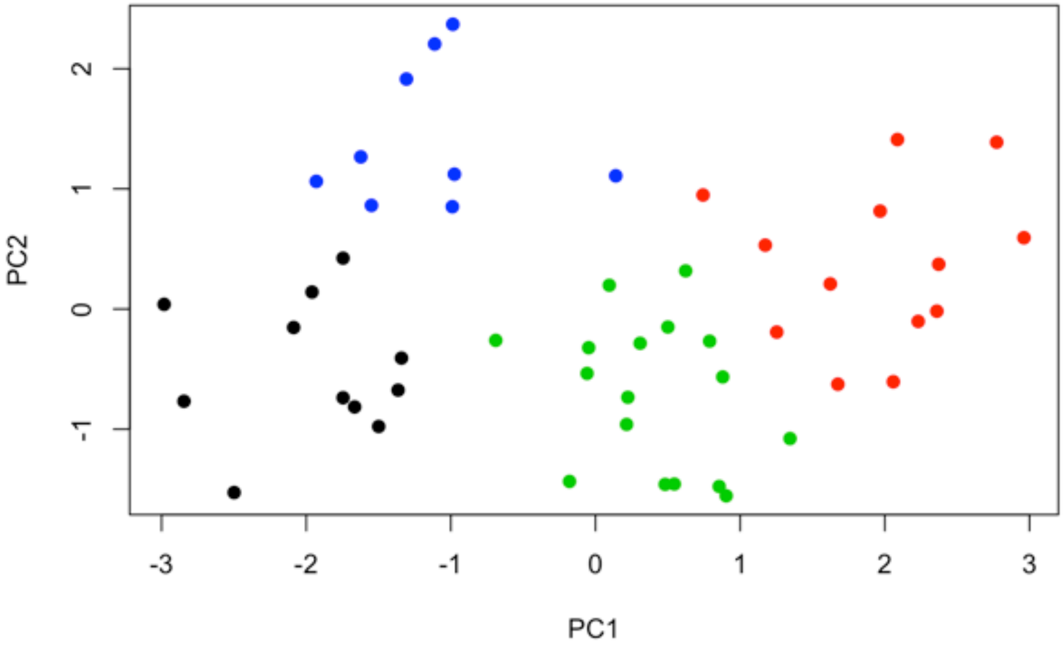


```
#clustering with data using only PC1 and PC2 and ideal k value
pca.kclusters<- kmeans(PCA.arr$x[,1:2], 4, iter.max=500, nstart=20)
```

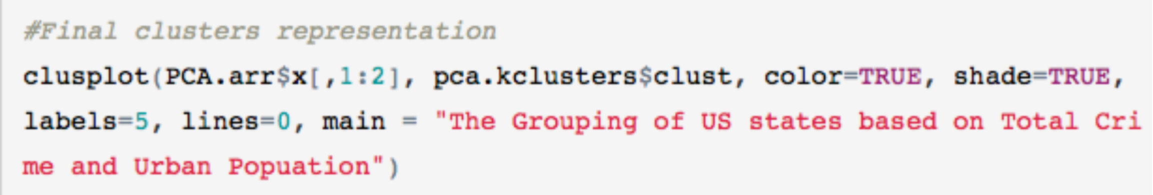
Plotting the above data points with different colors representing different clusters



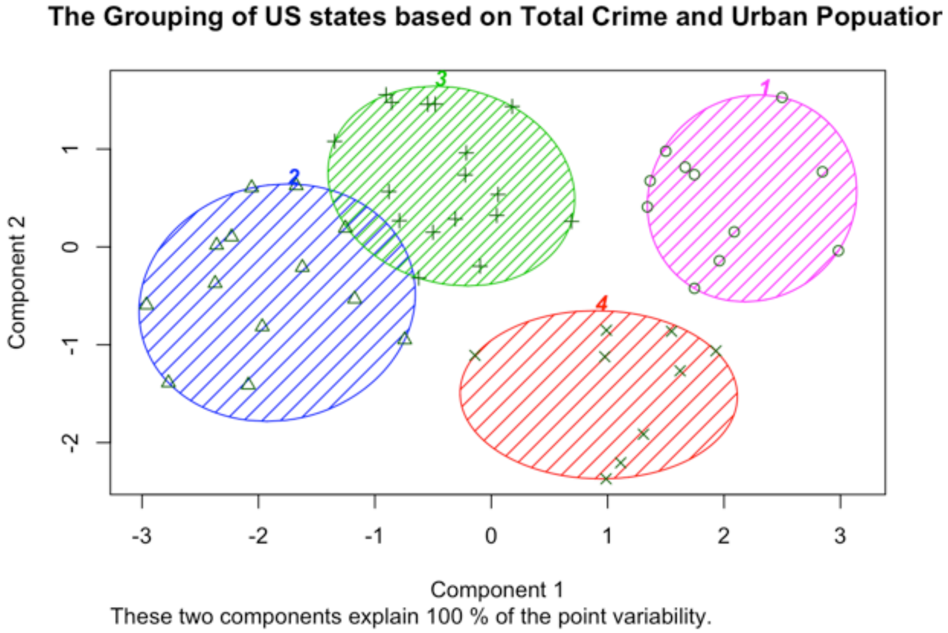
```
#kmeans clusters with maximum variance represented by PCA
plot(PCA.arr$x[,1:2], col=pca.kclusters$clust, pch=19,xlab="PC1",ylab="
PC2")
```



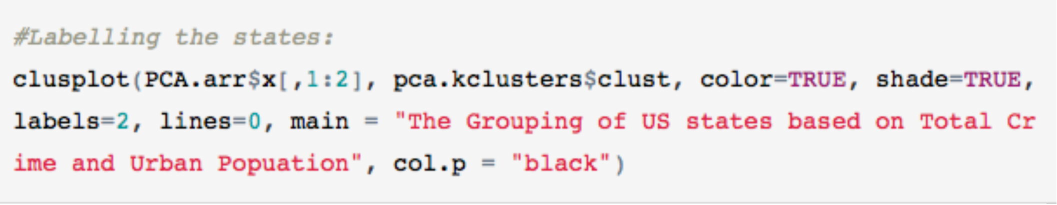
That looks a lot better and distinct when compared to the clustering of original data in the previous part.



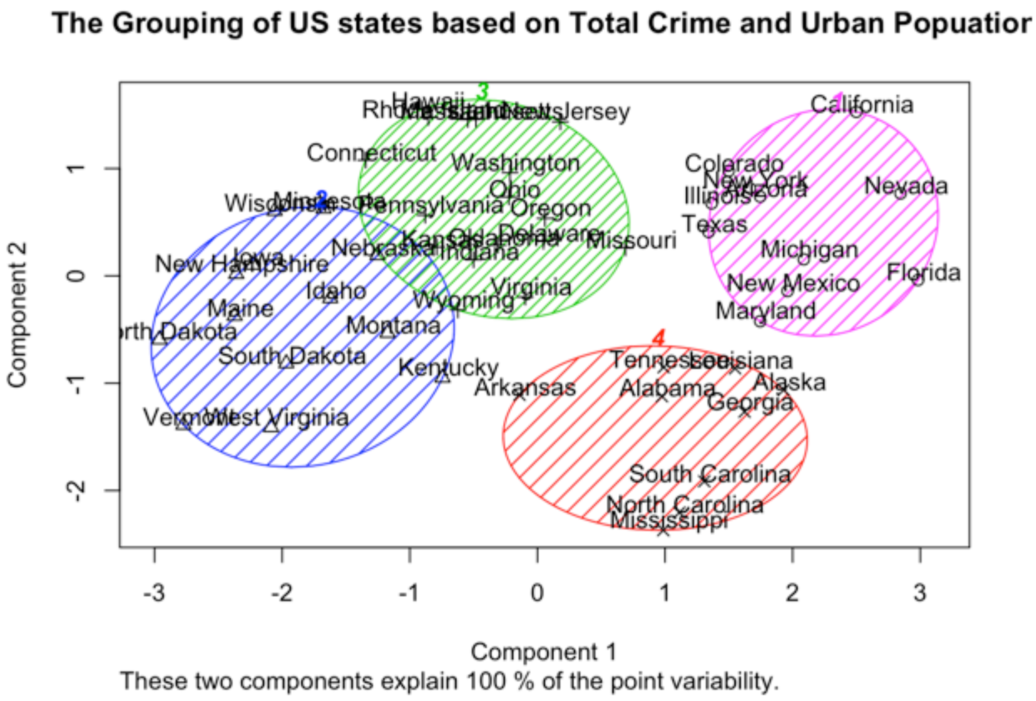
```
#Final clusters representation
clusplot(PCA.arr$x[,1:2], pca.kclusters$clust, color=TRUE, shade=TRUE,
labels=5, lines=0, main = "The Grouping of US states based on Total Cri
me and Urban Popuation")
```



Labeling the above points with state names will enable for good interpretations



```
#Labelling the states:
clusplot(PCA.arr$x[,1:2], pca.kclusters$clust, color=TRUE, shade=TRUE,
labels=2, lines=0, main = "The Grouping of US states based on Total Cr
ime and Urban Popuation", col.p = "black")
```



This clustering can be interpreted the exact same way as we interpreted the "BIPLOT"

Our X axis here is "Total Crime" and Y axis is "Urban Population" and the states fit exactly on this scale according to the biplot.

<u>Interpretations from the Graph</u>

1. Nevada, California, Michigan, New Mexico etc have high total crime while on the other side, West Virgiia, North Dakota, Marine etc have low total crime.
2. Within the 1st cluster, we can also say California is high in Urban Population along with high crime
3. Similarly, we can say New Jersy, Rhode ISland, Hawaii have a high Urban Population with Arkansas, Mississipi, North Carolina on the rural side.
4. New Hampshire is good on Urban population and very low on total crime.