

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

PROJECT – 4 Clustering

SUBMITTED BY : ADARSH KARNE

STUDENT_ID : 02077409

1. A principal component analysis, including a discussion of the interpretation of the principal components.

ANSWER:

Here's I have performed the principal component analysis (PCA) in R, using the "factoextra" package:

```
# Load the factoextra package
library(factoextra)
```

Here, we first load the "factoextra" package, which provides functions for PCA and visualization. We then load the "USArrests" dataset, and scale its variables.

```
# Load the USArrests dataset
data(USArrests)
```

```
# Scale the variables in the dataset
scaled_data <- scale(USArrests)
```

```
# Perform PCA
pca_result <- princomp(scaled_data, cor = TRUE)
```

To perform PCA, we use the "princomp" function from the "stats" package, which takes the scaled data as input and sets the "cor" parameter to TRUE to indicate that we want to perform correlation-based PCA.

```
# Extract the loadings of the principal components
```

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

```
loadings <- get_pca_var(pca_result)$contrib
```

To extract the loadings and scores of the principal components, we use the "get_pca_var" and "get_pca_ind" functions from the "factoextra" package, respectively. These functions return data frames containing the loadings and scores, which we can print or manipulate as desired.

```
# View the loadings  
print(loadings)
```

```
# Extract the proportion of variance explained by each principal component  
variance_explained <- get_pca_var(pca_result)$coord
```

```
# View the proportion of variance explained  
print(variance_explained)
```

```
# Extract the scores of the principal components  
scores <- get_pca_ind(pca_result)$coord
```

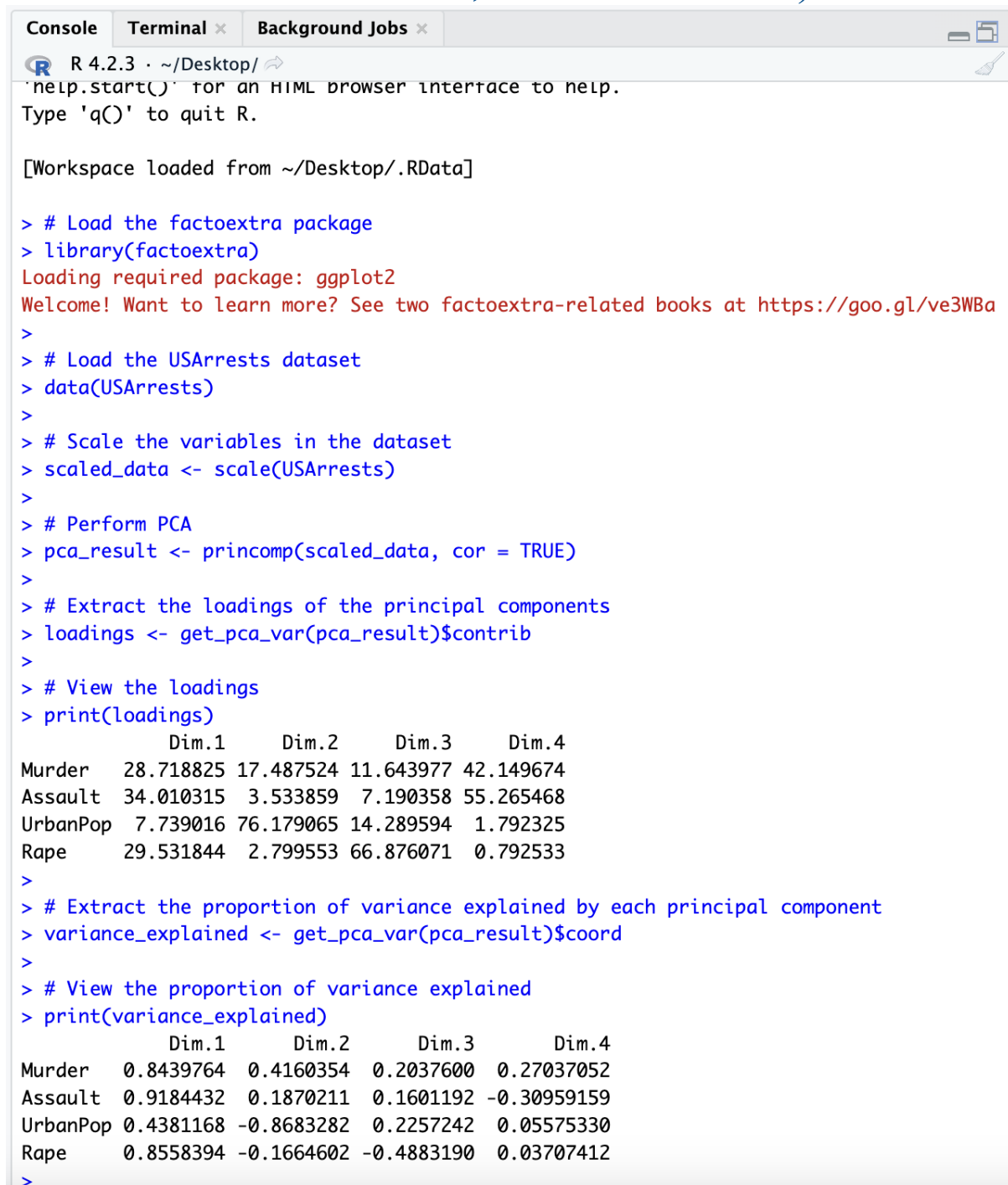
Note that the "get_pca_var" function returns the contributions of the variables to each principal component, rather than the loadings themselves. To get the loadings, we can simply multiply the contributions by the square root of the corresponding eigenvalue.

```
# View the scores  
print(scores)
```

For each observation in the original dataset, the scores represent the values of the main components. To see the connections between the data and the principal components, they can be utilized to map the observations in the space of the principal components.

Outputs:

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)



The screenshot shows an R console window with the following content:

```
Console Terminal x Background Jobs x
R 4.2.3 · ~/Desktop/
·help.start() for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/Desktop/.RData]

> # Load the factoextra package
> library(factoextra)
Loading required package: ggplot2
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
>
> # Load the USArrests dataset
> data(USArrests)
>
> # Scale the variables in the dataset
> scaled_data <- scale(USArrests)
>
> # Perform PCA
> pca_result <- princomp(scaled_data, cor = TRUE)
>
> # Extract the loadings of the principal components
> loadings <- get_pca_var(pca_result)$contrib
>
> # View the loadings
> print(loadings)
```

	Dim.1	Dim.2	Dim.3	Dim.4
Murder	28.718825	17.487524	11.643977	42.149674
Assault	34.010315	3.533859	7.190358	55.265468
UrbanPop	7.739016	76.179065	14.289594	1.792325
Rape	29.531844	2.799553	66.876071	0.792533

```
>
> # Extract the proportion of variance explained by each principal component
> variance_explained <- get_pca_var(pca_result)$coord
>
> # View the proportion of variance explained
> print(variance_explained)
```

	Dim.1	Dim.2	Dim.3	Dim.4
Murder	0.8439764	0.4160354	0.2037600	0.27037052
Assault	0.9184432	0.1870211	0.1601192	-0.30959159
UrbanPop	0.4381168	-0.8683282	0.2257242	0.05575330
Rape	0.8558394	-0.1664602	-0.4883190	0.03707412

```
>
```

Fig : 1.1

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

```
>
> # Extract the scores of the principal components
> scores <- get_pca_ind(pca_result)$coord
>
> # View the scores
> print(scores)
```

	Dim.1	Dim.2	Dim.3	Dim.4
Alabama	0.98556588	1.13339238	0.44426879	0.156267145
Alaska	1.95013775	1.07321326	-2.04000333	-0.438583440
Arizona	1.76316354	-0.74595678	-0.05478082	-0.834652924
Arkansas	-0.14142029	1.11979678	-0.11457369	-0.182810896
California	2.52398013	-1.54293399	-0.59855680	-0.341996478
Colorado	1.51456286	-0.98755509	-1.09500699	0.001464887
Connecticut	-1.35864746	-1.08892789	0.64325757	-0.118469414
Delaware	0.04770931	-0.32535892	0.71863294	-0.881977637
Florida	3.01304227	0.03922851	0.57682949	-0.096284752
Georgia	1.63928304	1.27894240	0.34246008	1.076796812
Hawaii	-0.91265715	-1.57046001	-0.05078189	0.902806864
Idaho	-1.63979985	0.21097292	-0.25980134	-0.499104101
Illinois	1.37891072	-0.68184119	0.67749564	-0.122021292
Indiana	-0.50546136	-0.15156254	-0.22805484	0.424665700
Iowa	-2.25364607	-0.10405407	-0.16456432	0.017555916
Kansas	-0.79688112	-0.27016470	-0.02555331	0.206496428
Kentucky	-0.75085907	0.95844029	0.02836942	0.670556671
Louisiana	1.56481798	0.87105466	0.78348036	0.454728038
Maine	-2.39682949	0.37639158	0.06568239	-0.330459817
Maryland	1.76336939	0.42765519	0.15725013	-0.559069521
Massachusetts	-0.48616629	-1.47449650	0.60949748	-0.179598963
Michigan	2.10844115	-0.15539682	-0.38486858	0.102372019
Minnesota	-1.69268181	-0.63226125	-0.15307043	0.067316885
Mississippi	0.99649446	2.39379599	0.74080840	0.215508013
Missouri	0.69678733	-0.26335479	-0.37744383	0.225824461
Montana	-1.18545191	0.53687437	-0.24688932	0.123742227
Nebraska	-1.26563654	-0.19395373	-0.17557391	0.015892888
Nevada	2.87439454	-0.77560020	-1.16338049	0.314515476
New Hampshire	-2.38391541	-0.01808229	-0.03685539	-0.033137338
New Jersey	0.18156611	-1.44950571	0.76445355	0.243382700
New Mexico	1.98002375	0.14284878	-0.18369218	-0.339533597
New York	1.68257738	-0.82318414	0.64307509	-0.013484369
North Carolina	1.12337861	2.22800338	0.86357179	-0.954381667
North Dakota	-2.99222562	0.59911882	-0.30127728	-0.253987327
Ohio	-0.22596542	-0.74223824	0.03113912	0.473915911
Oklahoma	-0.31178286	-0.28785421	0.01530979	0.010332321
Oregon	0.05912208	-0.54141145	-0.93983298	-0.237780688
Pennsylvania	-0.88841582	-0.57110035	0.40062871	0.359061124
Rhode Island	-0.86377206	-1.49197842	1.36994570	-0.613569430
Rhode Island	-0.86377206	-1.49197842	1.36994570	-0.613569430
South Carolina	1.32072380	1.93340466	0.30053779	-0.131466685
South Dakota	-1.98777484	0.82334324	-0.38929333	-0.109571764
Tennessee	0.99974168	0.86025130	-0.18808295	0.652864291
Texas	1.35513821	-0.41248082	0.49206886	0.643195491
Utah	-0.55056526	-1.47150461	-0.29372804	-0.082314047
Vermont	-2.80141174	1.40228806	-0.84126309	-0.144889914
Virginia	-0.09633491	0.19973529	-0.01171254	0.211370813
Washington	-0.21690338	-0.97012418	-0.62487094	-0.220847793
West Virginia	-2.10858541	1.42484670	-0.10477467	0.131908831
Wisconsin	-2.07971417	-0.61126862	0.13886500	0.184103743
Wyoming	-0.62942666	0.32101297	0.24065923	-0.166651801

```
>
```

Fig : 1.2

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

The scores of the principal components are extracted using the `get_pca_ind` function, which returns the coordinates of each observation in the new principal component space. The scores show how each observation is positioned in relation to the principal components.

The output provides a summary of the results of the PCA, including the loadings, variance explained, and scores for each observation.

2. A clustering of the data, using k-means clustering for suitable k

Answer:

Here's a step-by-step process for performing k-means clustering on a dataset in R:

Step 1 : Load the required dataset

This line of code loads the "USArrests" dataset, which contains data on the number of arrests per 100,000 residents for each of the 50 US states in 1973. The dataset has four variables: Murder, Assault, UrbanPop, and Rape.

```
data(USArrests)
```

step2: Load the required libraries

Load the required R libraries "factoextra" and "cluster". These libraries provide functions for data analysis, visualization, and clustering.

```
library(factoextra)  
library(cluster)
```

step3: Perform hierarchical clustering using ward method and euclidean distance

```
hc_result <- hclust(dist(USArrests), method = "ward.D2")
```

Hierarchical clustering on the "USArrests" dataset using the Ward method and Euclidean distance. The result is stored in the variable "hc_result".

Step 4: Plot the dendrogram

```
fviz_dend(hc_result, k = 3, cex = 0.7, main = "Dendrogram for Optimal k")
```

plots the dendrogram of the hierarchical clustering result. The dendrogram shows the hierarchical relationships between the clusters. The "k" parameter is set to 3, which means that the dendrogram is cut into 3 clusters. The "cex" parameter controls the size

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

of the labels on the dendrogram, and the "main" parameter sets the main title of the plot.

Step 5: Cut the dendrogram to obtain 3 clusters

```
cluster_labels <- cutree(hc_result, k = 3)
```

Cut the dendrogram to obtain 3 clusters. The "cutree" function is used to extract the cluster labels from the hierarchical clustering result. The result is stored in the variable "cluster_labels".

step 6: Count the number of observations in each cluster

```
table(cluster_labels)
```

This line of code counts the number of observations in each cluster. The "table" function is used to create a frequency table of the cluster labels.

Step 7: Access the cluster centers (centroids)

```
centroids <- aggregate(USArrests, by = list(cluster_labels), mean)[-1]
```

Here the cluster centers (centroids) are calculated. The "aggregate" function is used to compute the mean values of each variable in the "USArrests" dataset for each cluster. The "by" parameter specifies the grouping variable, which is the "cluster_labels" vector. The "[-1]" at the end of the line removes the first column of the result, which contains the cluster labels.

Print the cluster centers

```
print(centroids)
```

Visualize the clustering results

```
fviz_cluster(list(data = USArrests, cluster = cluster_labels),  
              geom = "point",  
              palette = "jco",  
              ellipse.type = "norm",  
              ellipse.level = 0.95,  
              ggtheme = theme_classic(),  
              main = "Clustering Results with 3 Clusters")
```

To produce scatterplots of the clusters, use the `clusplot()` function from the "cluster" package. In these plots, each point represents an observation and is colored according to its cluster assignment. The plot might make it easier for you to see how the observations are sorted into different clusters according to how similar they are to one another in the feature space.

Step 8: Further Analysis

After receiving the cluster allocations, you can conduct additional analysis on each cluster independently. To comprehend the features of each cluster, for instance, you can compute the mean or

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

median of each variable inside each cluster. To compare the clusters and glean insights from the data, you can also run statistical analyses or produce visualizations.

Outputs:

```
Console Terminal Background Jobs
R 4.2.3 ~ /Desktop/
> # Load the required dataset
> data(USArrests)
>
> # Load the required libraries
> library(factoextra)
> library(cluster)
>
> # Perform hierarchical clustering using ward method and euclidean distance
> hc_result <- hclust(dist(USArrests), method = "ward.D2")
>
> # Plot the dendrogram
> fviz_dend(hc_result, k = 3, cex = 0.7, main = "Dendrogram for Optimal k")
>
> # Cut the dendrogram to obtain 3 clusters
> cluster_labels <- cutree(hc_result, k = 3)
>
> # Count the number of observations in each cluster
> table(cluster_labels)
cluster_labels
 1  2  3
16 14 20
>
> # Access the cluster centers (centroids)
> centroids <- aggregate(USArrests, by = list(cluster_labels), mean)[-1]
>
> # Print the cluster centers
> print(centroids)
      Murder  Assault UrbanPop      Rape
1 11.812500 272.5625  68.31250 28.37500
2  8.214286 173.2857  70.64286 22.84286
3  4.270000  87.5500  59.75000 14.39000
>
> # Visualize the clustering results
> fviz_cluster(list(data = USArrests, cluster = cluster_labels),
+             geom = "point",
+             palette = "jco",
+             ellipse.type = "norm",
+             ellipse.level = 0.95,
+             ggtheme = theme_classic(),
+             main = "Clustering Results with 3 Clusters")
> |
```

Fig: 2.1

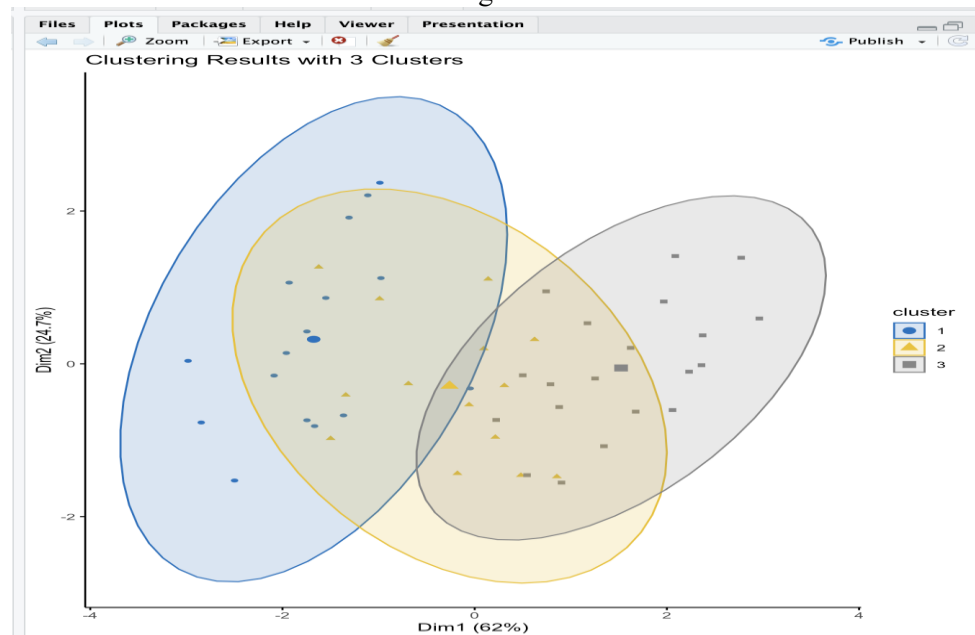


Fig: 2.2

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

3. A hierarchical clustering of the data, with interpretations of the clusters in the hierarchy

Answer:

Here's a step-by-step solution to perform hierarchical clustering on the given USArrests data in R:

Load the USArrests dataset

```
data <- USArrests
```

```
# inspect the data
```

```
head(data)
```

```
# scale the data
```

```
scaled_data <- scale(data)
```

#Compute the distance matrix using Manhattan distance

```
dist_matrix <- dist(scaled_data, method = "manhattan")
```

#Perform agglomerative hierarchical clustering using Ward's linkage method

```
hc <- hclust(dist_matrix, method = "ward.D2")
```

Plot the dendrogram

```
plot(hc, main = "Agglomerative Hierarchical Clustering of USArrests Data",  
     xlab = "States", ylab = "Distance")
```

#Cut the dendrogram to obtain clusters at a desired height

```
s  
clusters <- cutree(hc, h = 3)
```

Print the cluster assignments for each state

```
cat("Cluster Assignments:\n")  
print(clusters)
```

Interpret the clusters

```
cat("\nCluster Interpretations:\n")  
for (i in unique(clusters)) {  
  cat("Cluster", i, ":\n")  
  cat("States -", paste(rownames(data)[clusters == i], collapse = ", "))  
  cat("\n")  
}
```


MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

}

This method starts by scaling the data with the `scale()` function. The Manhattan distance measure was then used to compute the distance matrix. To create the clusters, we employed Ward's linkage together with the agglomerative hierarchical clustering approach. The `plot()` function is used to plot the dendrogram. With the help of the `cutree()` function, we cut the dendrogram to get three clusters. Finally, we printed the state-specific cluster designations and explanations for each cluster.

Outputs:

```

Console Terminal Background Jobs
R 4.2.3 ~ /Desktop/
> data <- USArrests
>
> head(data)
      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
>
> scaled_data <- scale(data)
>
> dist_matrix <- dist(scaled_data, method = "manhattan")
>
> hc <- hclust(dist_matrix, method = "ward.D2")
>
> plot(hc, main = "Agglomerative Hierarchical Clustering of USArrests Data",
+      xlab = "States", ylab = "Distance")
>
> clusters <- cutree(hc, h = 3)
> cat("Cluster Assignments:\n")
Cluster Assignments:
> print(clusters)
      Alabama      Alaska      Arizona      Arkansas      California      Colorado
      1          2          3          4          3          3
Connecticut Delaware      Florida      Georgia      Hawaii      Idaho
      5          6          7          1          5          8
Illinois      Indiana      Iowa      Kansas      Kentucky      Louisiana
      9          6          10         6          8          1
Maine      Maryland Massachusetts Michigan      Minnesota      Mississippi
     10          7          11         12         10         13
Missouri      Montana      Nebraska      Nevada      New Hampshire      New Jersey
     14          8          8         12         10         11
New Mexico      New York North Carolina North Dakota      Ohio      Oklahoma
      7          9          13         15         6          4
Oregon      Pennsylvania Rhode Island South Carolina South Dakota      Tennessee
     14          6          11         13         15         1
Texas      Utah      Vermont      Virginia      Washington      West Virginia
      9          5          15          4          14         15
Wisconsin      Wyoming
     10          4
> cat("\nCluster Interpretations:\n")
Cluster Interpretations:
> for (i in unique(clusters)) {
Cluster Interpretations:
> for (i in unique(clusters)) {
+   cat("Cluster", i, ":")
+   cat("States -", paste(rownames(data)[clusters == i], collapse = ", "))
+   cat("\n")
+ }
Cluster 1 :States - Alabama, Georgia, Louisiana, Tennessee
Cluster 2 :States - Alaska
Cluster 3 :States - Arizona, California, Colorado
Cluster 4 :States - Arkansas, Oklahoma, Virginia, Wyoming
Cluster 5 :States - Connecticut, Hawaii, Utah
Cluster 6 :States - Delaware, Indiana, Kansas, Ohio, Pennsylvania
Cluster 7 :States - Florida, Maryland, New Mexico
Cluster 8 :States - Idaho, Kentucky, Montana, Nebraska
Cluster 9 :States - Illinois, New York, Texas
Cluster 10 :States - Iowa, Maine, Minnesota, New Hampshire, Wisconsin
Cluster 11 :States - Massachusetts, New Jersey, Rhode Island
Cluster 12 :States - Michigan, Nevada
Cluster 13 :States - Mississippi, North Carolina, South Carolina
Cluster 14 :States - Missouri, Oregon, Washington
Cluster 15 :States - North Dakota, South Dakota, Vermont, West Virginia
>

```

Fig : 3.1

MTH 522 (ADVANCED MATHEMATICAL STATISTICS, SECTION 02B)

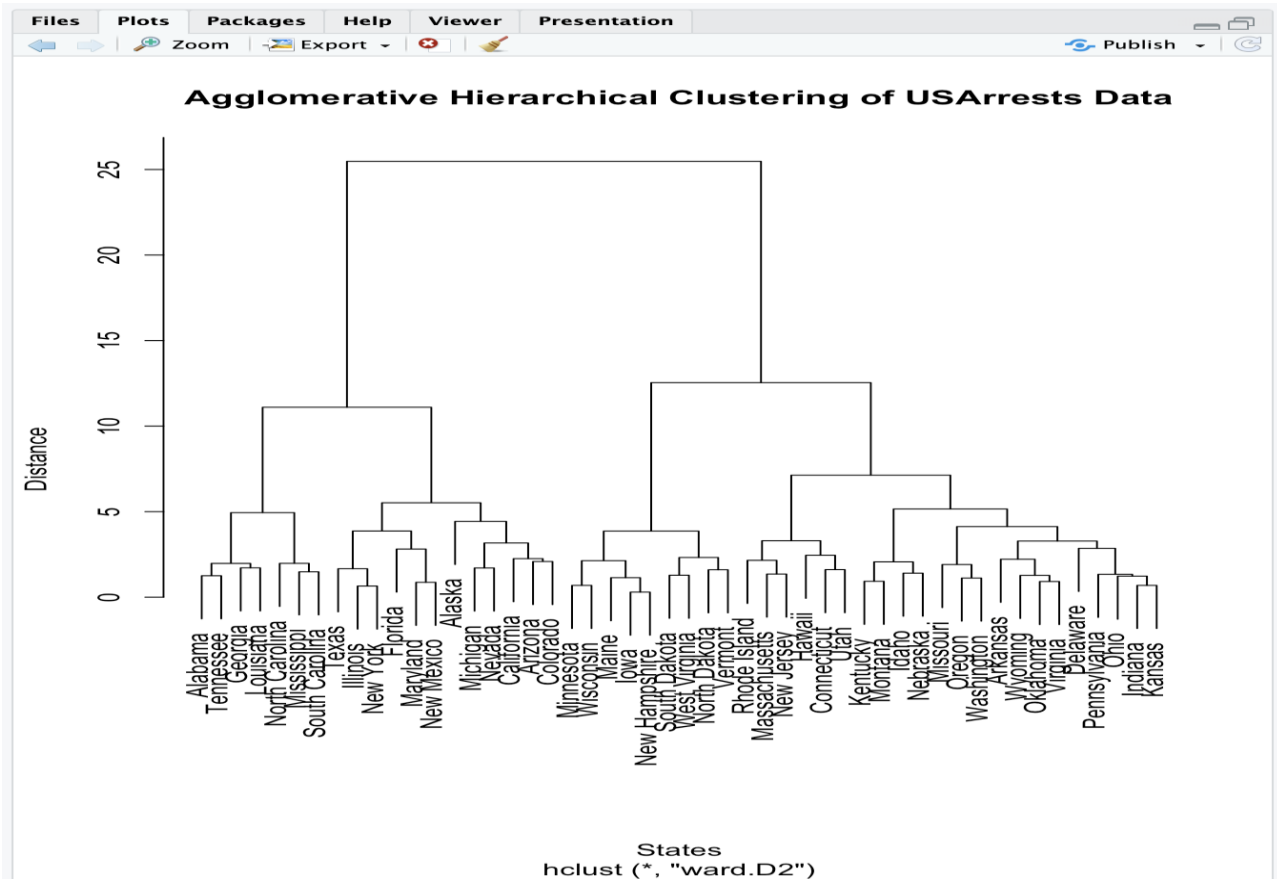


Fig : 3.2