University of Jeddah
College of Computer Science and Engineering
Department of Information Systems & Technology

Violent Crime Rates by US State

Name: Rahaf Khalid Alhuzali

University of Jeddah

College of Computer Science and Engineering

Department of Information Systems & Technology

CCDS211 – Introduction to Data Science

Lab Project (10 marks)

Introduction:

The Problem:

Because of the large number of reported crimes committed in the United States in several states

It has become noticeably abundant, and there is no place without killing and assault, because of the lack of pieces of information, with not explaining the reason for many crimes in several states in terms of murder and assaults, finding relationships within critical data.

Abstract:

The aim of this social study efficiently is to find relationships within data. Know the rates of violent crimes through murders and assaults.

Dataset:

This data set contains statistics, in arrests per 100,000 residents for assault and murder, in each of the 200 US states. Also given is the percent of the population living in urban areas.

A data frame with 200 observations on 3 variables.

City is String .

Murder is numeric and Murder arrests (per 100,000)
Assault is numeric and Assault arrests (per 100,000)

The Colum it is :

- City
- Murder
- Assault

References: McNeil, D. R. (1977) Interactive Data Analysis. New York: Wiley.

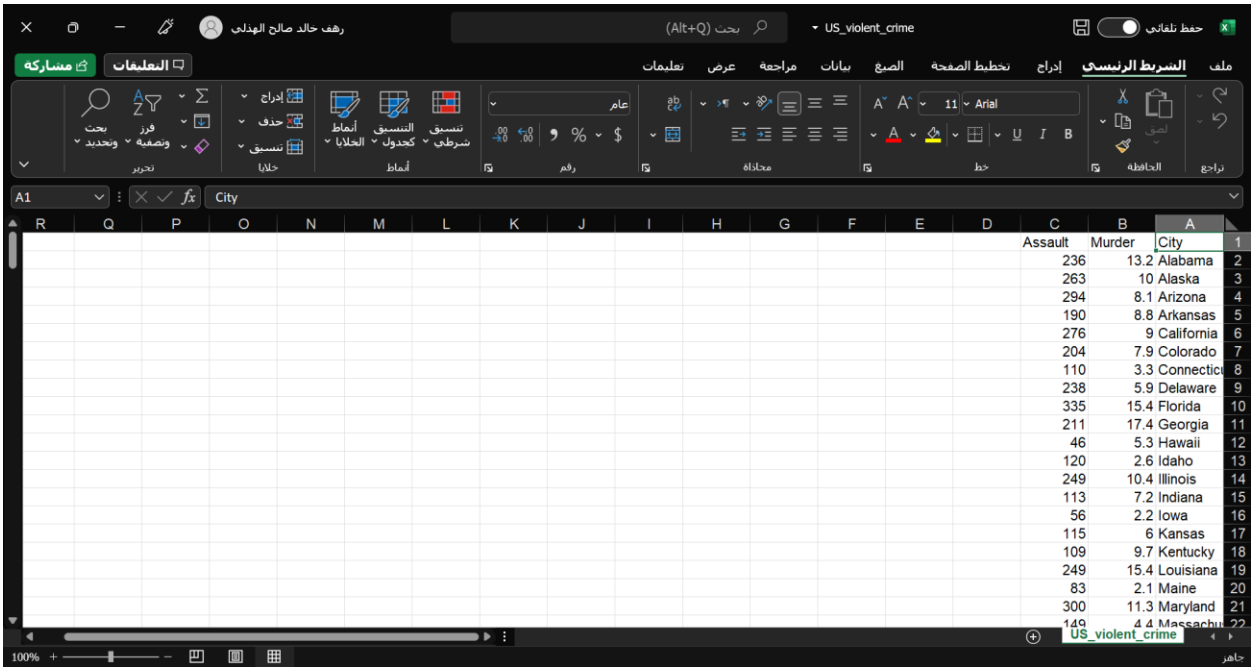
Analysis:

This data set contains statistics, in arrests per 100,000 residents for assault and murder, in each of the 200 US states. Also given is the percent of the population living in urban areas.

A data frame with 200 observations on 3 variables.

City is String .

Murder is numeric and Murder arrests (per 100,000)
Assault is numeric and Assault arrests (per 100,000)



Assault	Murder	City	
236	13.2	Alabama	2
263	10	Alaska	3
294	8.1	Arizona	4
190	8.8	Arkansas	5
276	9	California	6
204	7.9	Colorado	7
110	3.3	Connectic	8
238	5.9	Delaware	9
335	15.4	Florida	10
211	17.4	Georgia	11
46	5.3	Hawaii	12
120	2.6	Idaho	13
249	10.4	Illinois	14
113	7.2	Indiana	15
56	2.2	Iowa	16
115	6	Kansas	17
109	9.7	Kentucky	18
249	15.4	Louisiana	19
83	2.1	Maine	20
300	11.3	Maryland	21
149	4.4	Massachu	22

University of Jeddah

College of Computer Science and Engineering

Department of Information Systems & Technology

Applying Machine Learning: Clustering

Getting Data

```
Console   Terminal   Jobs
R 4.1.2 · ~/R/project/

> library("ggfortify")
> library(tidyverse)  # data manipulation
-- Attaching packages --------------------------------------- tidyverse 1.3.1 --
v tibble  3.1.6     v purrr   0.3.4
v tidyr   1.1.4     v stringr 1.4.0
v readr   2.1.0     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
> library(cluster)    # clustering algorithms
> library(factoextra) # clustering algorithms & visualization
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
> getwd()
[1] "C:/Users/Dell1/Documents"
> setwd("C:/Users/Dell1/Documents/R/project")
> list.files()
[1] "US_violent_crime.csv"  "US_violent_crime.xlsx"
> project=read.csv("US_violent_crime.csv")
> project=na.omit(project)
> str(project)
'data.frame':   212 obs. of  3 variables:
 $ City   : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
 $ Murder : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
 $ Assault: num  236 263 294 190 276 204 110 238 335 211 ...
> head(project)
        City Murder Assault
1    Alabama   13.2     236
2     Alaska   10.0     263
3    Arizona    8.1     294
4   Arkansas    8.8     190
5 California    9.0     276
6   Colorado    7.9     204
```

Basically, I used a pairs plot will provide a scatter plot for all possible combinations

```
> pairs(project[2:3])
```

Normalize / Calculate distance matrix

```
> z <- project[,-c(1,1)]
> means <- apply(z,2,mean)
> sds <- apply(z,2,sd)
> nor <- scale(z,center=means,scale=sds)
> distance = dist(nor)
```

Hierarchical agglomerative clustering

```
> distance = dist(nor)
> project.hclust = hclust(distance)
> plot(project.hclust)
> plot(project.hclust,labels=project$City,main='Default from hclust')
> plot(project.hclust,hang=-1, labels=project$City,main='Default from hclust')
> project.hclust<-hclust(distance,method="average")
```

```
> aggregate(nor,list(member),mean)
  Group.1      Murder    Assault
1       1 -0.59768873  2.5129656
2       2 -0.06888764 -0.2958464
3       3  2.55489278 -0.2032900
> aggregate(project[,-c(1,1)],list(member),mean)
  Group.1   Murder   Assault
```

Hierarchical agglomerative clustering using "average" linkage

```
> project.hclust<-hclust(distance,method="average")
> plot(project.hclust,hang=-1)
```

Cluster membership

```
> member = cutree(project.hclust,3)
> table(member)
member
  1   2   3
 22 180  10
> member
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  1   1   1   1   1   1   2   1   1   1   2   2   1   2   2   2   2
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  1   2   1   2   1   2   1   1   2   2   1   2   2   1   1   1   2
 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
  2   2   2   2   2   1   2   1   1   2   2   2   2   2   2   2   2
 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
  3   2   2   3   3   2   2   2   2   2   2   2   2   2   2   2   2
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  2   2   2   2   2   2   2   2   2   3   2   2   2   2   2   2   2
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   2
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  1   1   1   1   1   1   2   1   1   1   2   2   1   2   2   2   2
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  1   2   1   2   1   2   1   1   2   2   1   2   2   1   1   1   2
 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
  2   2   2   2   2   1   2   1   1   2   2   2   2   2   2   2   2
 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
  3   2   2   3   3   2   2   2   2   2   2   2   2   2   2   2   2
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  2   2   2   2   2   2   2   2   2   3   2   2   2   2   2   2   2
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   2
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
  2   2   2   3   2   2   2   2   2   2   2   2   2   2   2   2   2
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
  2   2   2   2   2   2   2   2   2   2   3   2   2   3   3   2   2
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
  2   2   3   2   2   2   2   2   2   2   2   2   2   2   2   2   2
188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
205 206 207 208 209 210 211 212
  2   2   2   2   2   2   2   2
```

Characterizing clusters

```
> aggregate(nor,list(member),mean)
  Group.1      Murder    Assault
1       1 -0.59768873  2.5129656
2       2 -0.06888764 -0.2958464
3       3  2.55489278 -0.2032900
> aggregate(project[,-c(1,1)],list(member),mean)
  Group.1   Murder   Assault
1       1 11.72727 251.50000
2       2 16.87722  54.90167
3       3 42.43000  61.38000
```

Silhouette Plot

```
2        2 16.87722  54.90167
3        3 42.43000  61.38000
> library(cluster)
> plot(silhouette(cutree(project.hclust,10), distance))
>
```

Scree plot

Scree plot will allow us to see the variabilities in clusters, suppose if we increase the number of clusters within-group sum of squares will come down.

```
> wss <- (nrow(nor)-1)*sum(apply(nor,2,var))
> for (i in 2:20) wss[i] <- sum(kmeans(nor, centers=i)$withinss)
> plot(1:20, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
```

```
> avg_sil <- function(k) {
+   km.res <- kmeans(nor, centers = k, nstart = 25)
+   ss <- silhouette(km.res$cluster, dist(nor))
+   mean(ss[, 3])
+ }
```

K-means clustering

```
> kc<-kmeans(nor,3)
> kc
K-means clustering with 3 clusters of sizes 97, 85, 30

Cluster means:
      Murder    Assault
1 -0.6335164 -0.3083750
2  0.9916678 -0.4076118
3 -0.7613558  2.1519793

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  3   3   3   3   3   3   1   3   3   3   1   1   3   1   1   1   1
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  3   1   3   3   3   1   3   3   1   3   3   1   3   3   3   3   1
 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
```

```
Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  3   3   3   3   3   3   1   3   3   3   1   1   3   1   1   1   1
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  3   1   3   3   3   1   3   3   1   1   3   1   3   3   3   3   1
 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
  1   3   3   1   3   3   1   3   3   1   1   3   3   1   1   3   2
 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
  2   2   2   2   2   1   1   2   2   2   1   2   2   1   1   1   2
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  1   2   1   2   1   1   2   1   1   2   1   1   2   2   1   1   2
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  2   2   1   1   2   1   2   2   2   1   2   2   1   1   1   2   1
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  1   2   2   2   1   2   2   1   1   1   2   1   2   1   2   1   1
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
  2   1   1   2   1   1   2   2   1   1   2   2   2   1   1   2   1
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
  2   2   2   1   2   2   1   1   1   2   2   2   2   2   2   1   1
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
  2   2   2   1   2   2   1   1   1   2   1   2   1   2   1   1   2
171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
  1   1   2   1   1   2   2   1   1   2   2   2   1   1   2   1   2
188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
  2   2   1   2   2   1   1   1   1   2   2   2   1   1   2   1   1
205 206 207 208 209 210 211 212
  2   2   2   1   2   2   1   1

Within cluster sum of squares by cluster:
[1] 42.35058 53.79545 23.66721
 (between_SS / total_SS =  71.6 %)
```

```
R 4.1.2   ~/R/project/
  2   2   2   2   2   1   1   2   2   2   1   2   2   1   1   1   2
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  1   2   1   2   1   1   2   1   1   2   1   1   2   2   1   1   2
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  2   2   1   1   2   1   2   2   2   1   2   2   1   1   1   2   1
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  1   2   2   2   1   2   2   1   1   1   2   1   2   1   2   1   1
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
  2   1   1   2   1   1   2   2   1   1   2   2   2   1   1   2   1
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
  2   2   2   1   2   2   1   1   1   2   2   2   2   2   2   1   1
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
  2   2   2   1   2   2   1   1   1   2   1   2   1   2   1   1   2
171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
  1   1   2   1   1   2   2   1   1   2   2   2   1   1   2   1   2
188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
  2   2   1   2   2   1   1   1   1   2   2   2   1   1   2   1   1
205 206 207 208 209 210 211 212
  2   2   2   1   2   2   1   1

Within cluster sum of squares by cluster:
[1] 42.35058 53.79545 23.66721
 (between_SS / total_SS =  71.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

Cluster Mapping

```
R 4.1.2 · ~/R/project/
> ot<-nor
> datadistshortset<-dist(ot,method = "euclidean")
> hc1 <- hclust(datadistshortset, method = "complete" )
> pamvshortset <- pam(datadistshortset,4, diss = FALSE)
> clusplot(pamvshortset, shade = FALSE,labels=2,col.clus="blue",col.p="red",span=FALSE,main="Cluster Mapping",cex=1.2)
```

Cluster Analysis

```
> k2 <- kmeans(nor, centers = 2, nstart = 25)
> str(k2)
List of 9
 $ cluster     : Named int [1:212] 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "names")= chr [1:212] "1" "2" "3" "4" ...
 $ centers     : num [1:2, 1:2] -0.908 0.224 1.668 -0.412
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "1" "2"
  .. ..$ : chr [1:2] "Murder" "Assault"
 $ totss       : num 422
 $ withinss    : num [1:2] 51.4 181.7
 $ tot.withinss: num 233
 $ betweenss   : num 189
 $ size        : int [1:2] 42 170
 $ iter        : int 1
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
```

Optimal Clusters:

```
> fviz_nbclust(nor, kmeans, method = "wss")
```

Average Silhouette Method:

I Used The average silhouette approach measures the quality of a clustering. It determines how well each observation lies within its cluster.

```
> fviz_nbclust(nor, kmeans, method = "silhouette")
```

Gap Statistic Method:

I used the gap statistic to compare the total intra cluster variation for different values of k with their expected values under the null reference distribution of the data.

```
> # compute gap statistic
> set.seed(123)
> gap_stat <- clusGap(nor, FUN = kmeans, nstart = 25,
+                     K.max = 10, B = 50)
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
.................................................. 50
> # Print the result
> print(gap_stat, method = "firstmax")
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = nor, FUNcluster = kmeans, K.max = 10, B = 50, nstart = 25)
B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstmax'): 1
          logW   E.logW       gap     SE.sim
 [1,] 4.490161 4.891443 0.4012818 0.02791313
 [2,] 4.191047 4.535512 0.3444657 0.02349209
 [3,] 3.873402 4.345664 0.4722623 0.01965877
 [4,] 3.710750 4.172930 0.4621801 0.02433966
 [5,] 3.561039 4.039980 0.4789407 0.02189095
 [6,] 3.456945 3.931725 0.4747796 0.02298658
 [7,] 3.354487 3.847964 0.4934769 0.02189399
 [8,] 3.267765 3.771858 0.5040934 0.02144039
 [9,] 3.183247 3.701918 0.5186714 0.02201696
[10,] 3.124453 3.637372 0.5129192 0.02148391
>
```

```
> fviz_gap_stat(gap_stat)
> # Compute k-means clustering with k = 4
> set.seed(123)
> final <- kmeans(nor, 4, nstart = 25)
> print(final)
K-means clustering with 4 clusters of sizes 88, 49, 23, 52

Cluster means:
        Murder     Assault
1  0.02166337 -0.46487958
2  1.42295189 -0.36975467
3 -0.63486487  2.46482447
```

K-means:5

```
> k3 <- kmeans(nor, centers = 3, nstart = 25)
> k4 <- kmeans(nor, centers = 4, nstart = 25)
> k5 <- kmeans(nor, centers = 5, nstart = 25)
> # plots to compare
> p1 <- fviz_cluster(k2, geom = "point", data = nor) + ggtitle("k =
 2")
> p2 <- fviz_cluster(k3, geom = "point",  data = nor) + ggtitle("k =
 3")
> p3 <- fviz_cluster(k4, geom = "point",  data = nor) + ggtitle("k =
 4")
> p4 <- fviz_cluster(k5, geom = "point",  data = nor) + ggtitle("k =
 5")
> grid.arrange(p1, p2, p3, p4, nrow = 2)
>
```

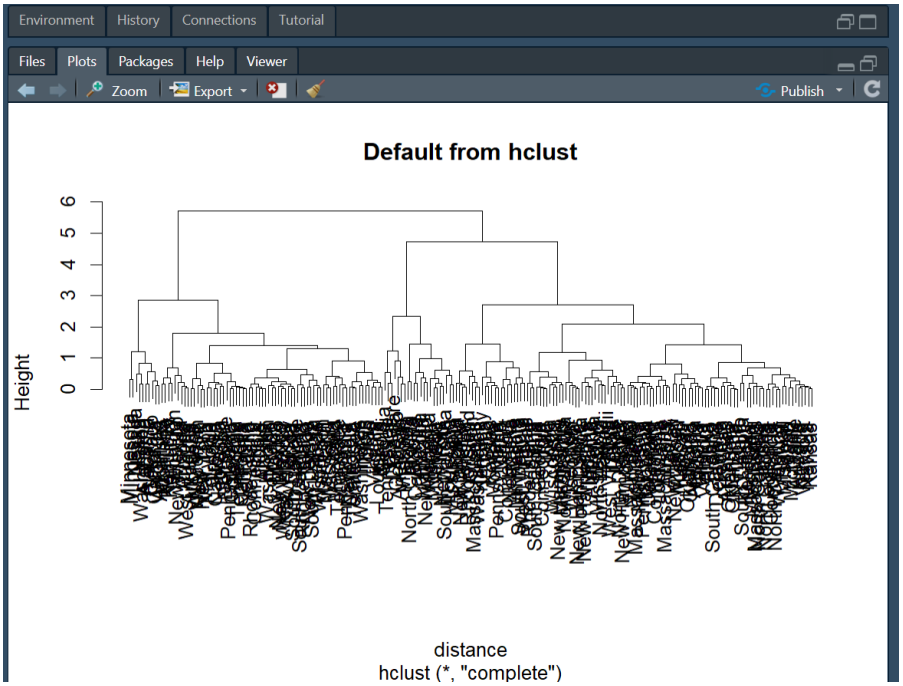
```
R 4.1.2 · ~/R/project/
 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
  2   2   1   2   2   4   1   2   2   1   1   1   1   4   1   1   1
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  4   2   1   2   1   1   2   1   1   2   4   1   2   2   1   4   1
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  1   2   1   4   1   1   2   2   1   1   1   2   4   4   1   2   4
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  1   2   2   1   1   1   1   4   1   1   1   4   2   1   2   1   1
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
  2   1   1   2   4   1   2   2   1   4   1   1   2   1   4   1   1
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
  2   2   1   4   1   2   4   4   1   1   2   2   1   2   2   1   1
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
  2   2   1   1   1   1   1   1   1   1   4   2   1   2   1   1   2
171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
  1   1   2   4   1   2   2   1   4   1   1   2   1   4   1   1   2
188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
  2   1   1   1   2   4   1   1   4   1   1   2   1   4   1   1   1
205 206 207 208 209 210 211 212
  2   2   1   1   1   2   4   1

Within cluster sum of squares by cluster:
[1] 22.55579 27.43209 12.21981 23.16592
 (between_SS / total_SS =  79.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
> fviz_cluster(final, data = nor)
>
```

```
> # Compute k-means clustering with k = 4
> set.seed(123)
> final <- kmeans(nor, 4, nstart = 25)
> print(final)
K-means clustering with 4 clusters of sizes 88, 49, 23, 52

Cluster means:
       Murder     Assault
1  0.02166337 -0.46487958
2  1.42295189 -0.36975467
3 -0.63486487  2.46482447
4 -1.09671397  0.04493114

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  3   3   3   3   3   3   4   3   3   3   4   4   3   4   4   4   4
 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  3   4   3   4   3   4   3   3   4   4   3   4   4   3   3   3   4
 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
  4   4   4   4   3   3   4   3   3   4   4   4   4   4   4   4   1
 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
  2   2   1   2   2   4   1   2   2   1   1   1   1   4   1   1   1
 69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
  4   2   1   2   1   1   2   1   1   2   4   1   2   2   1   4   1
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
  1   2   1   4   1   1   2   2   1   1   1   2   4   4   1   2   4
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
  1   2   2   1   1   1   1   4   1   1   1   4   2   1   2   1   1
120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
  2   1   1   2   4   1   2   2   1   4   1   1   2   1   4   1   1
137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
```

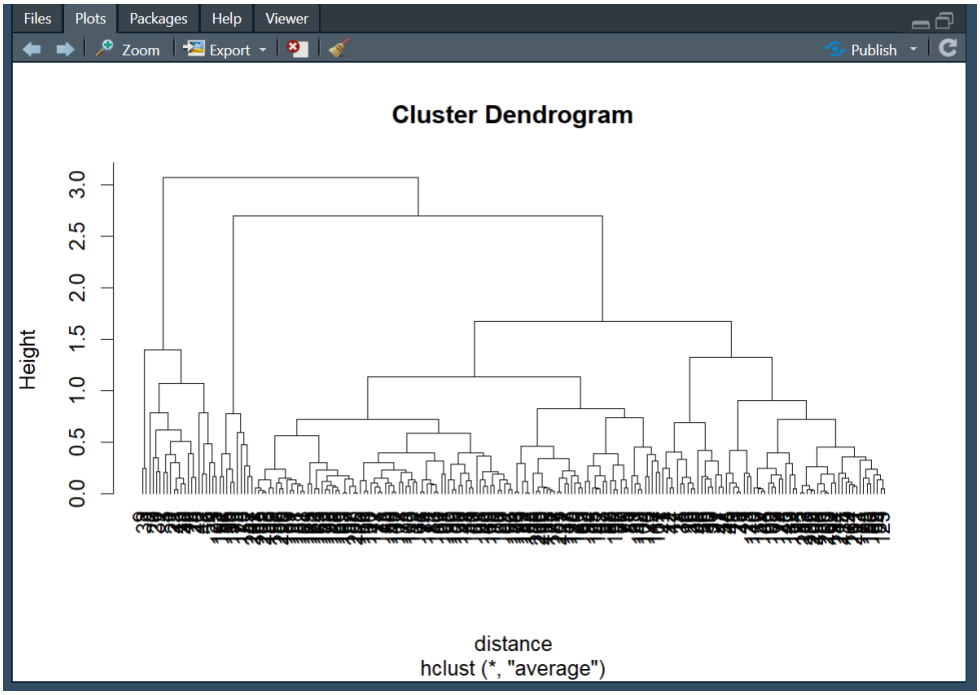
Plot Results:

Basically, I used a pairs plot will provide a scatter plot for all possible combinations

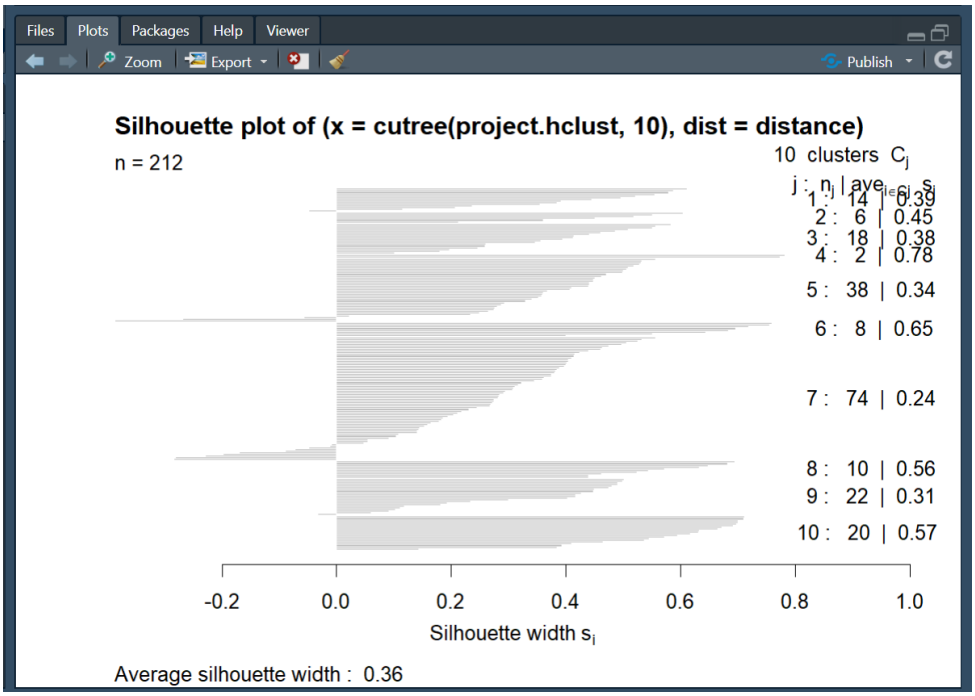
Hierarchical agglomerative clustering

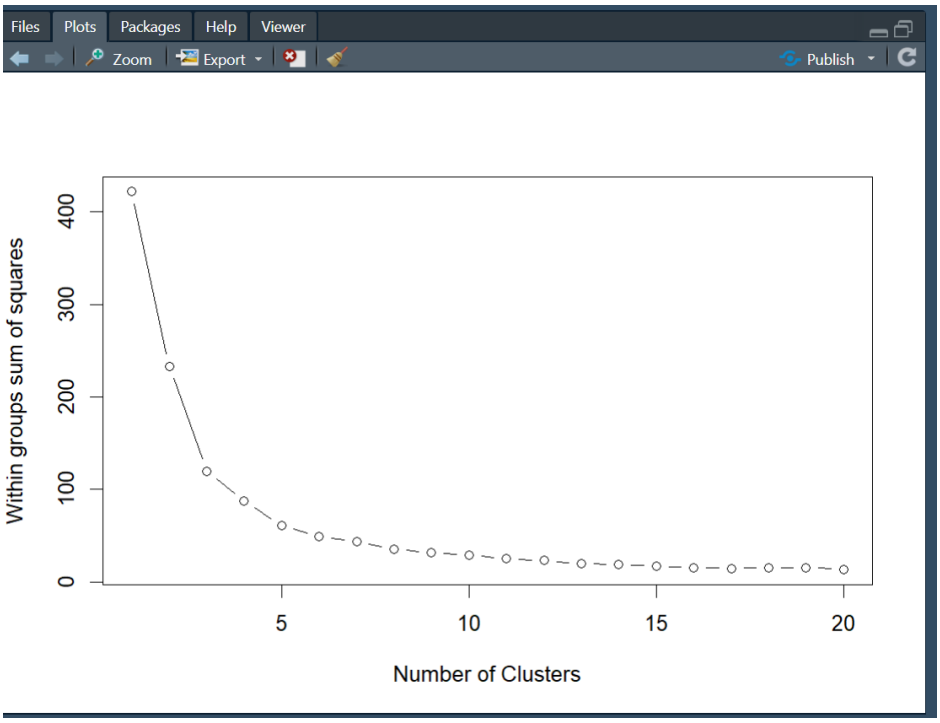


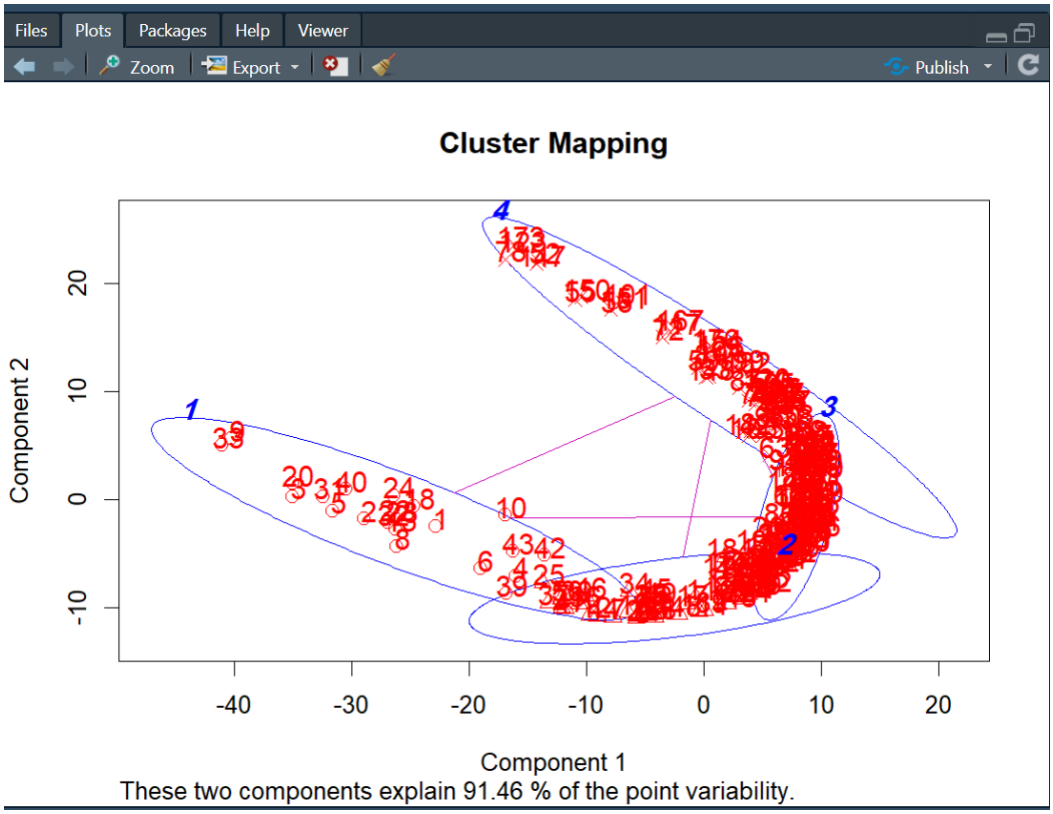
Hierarchical agglomerative clustering using "average" linkage

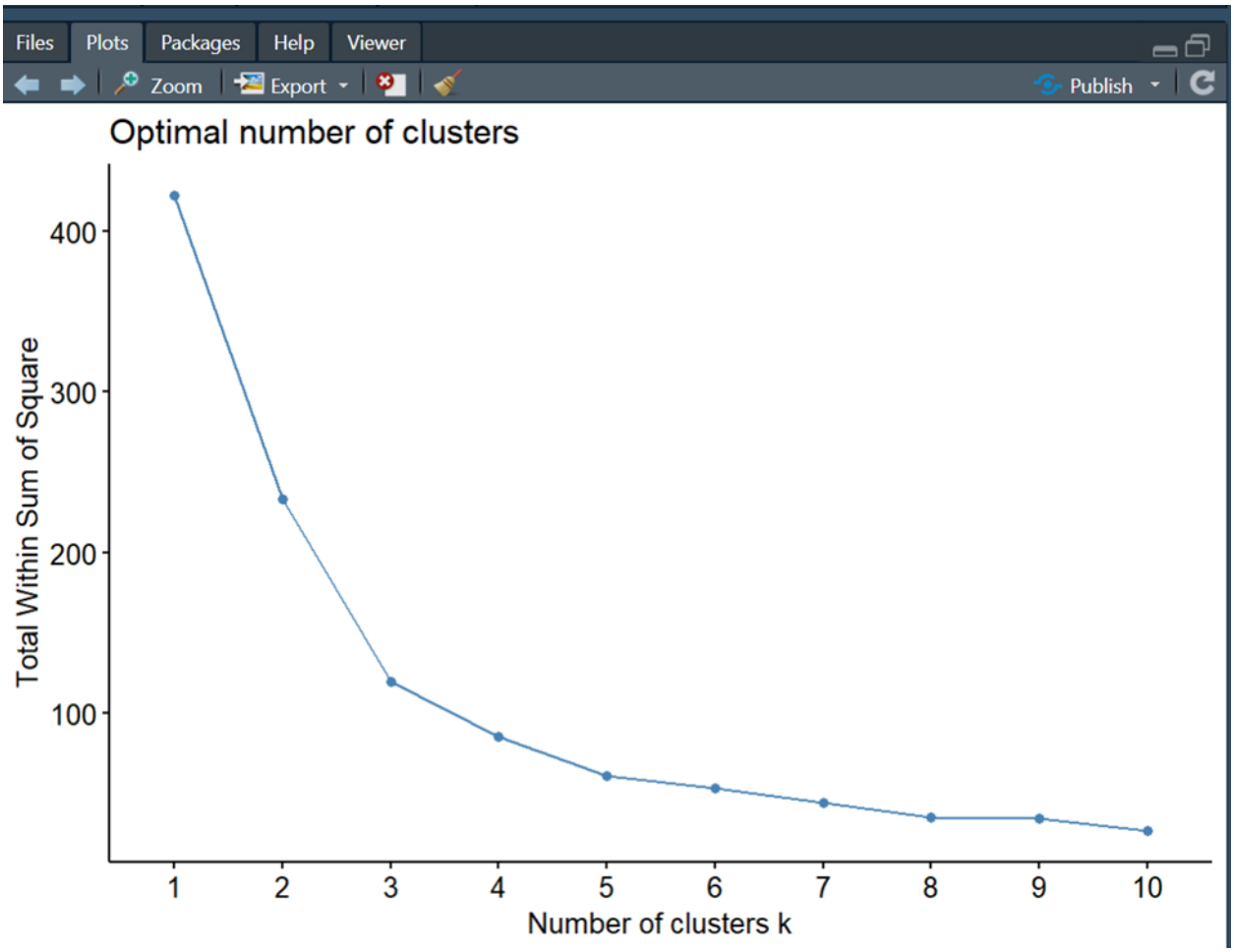


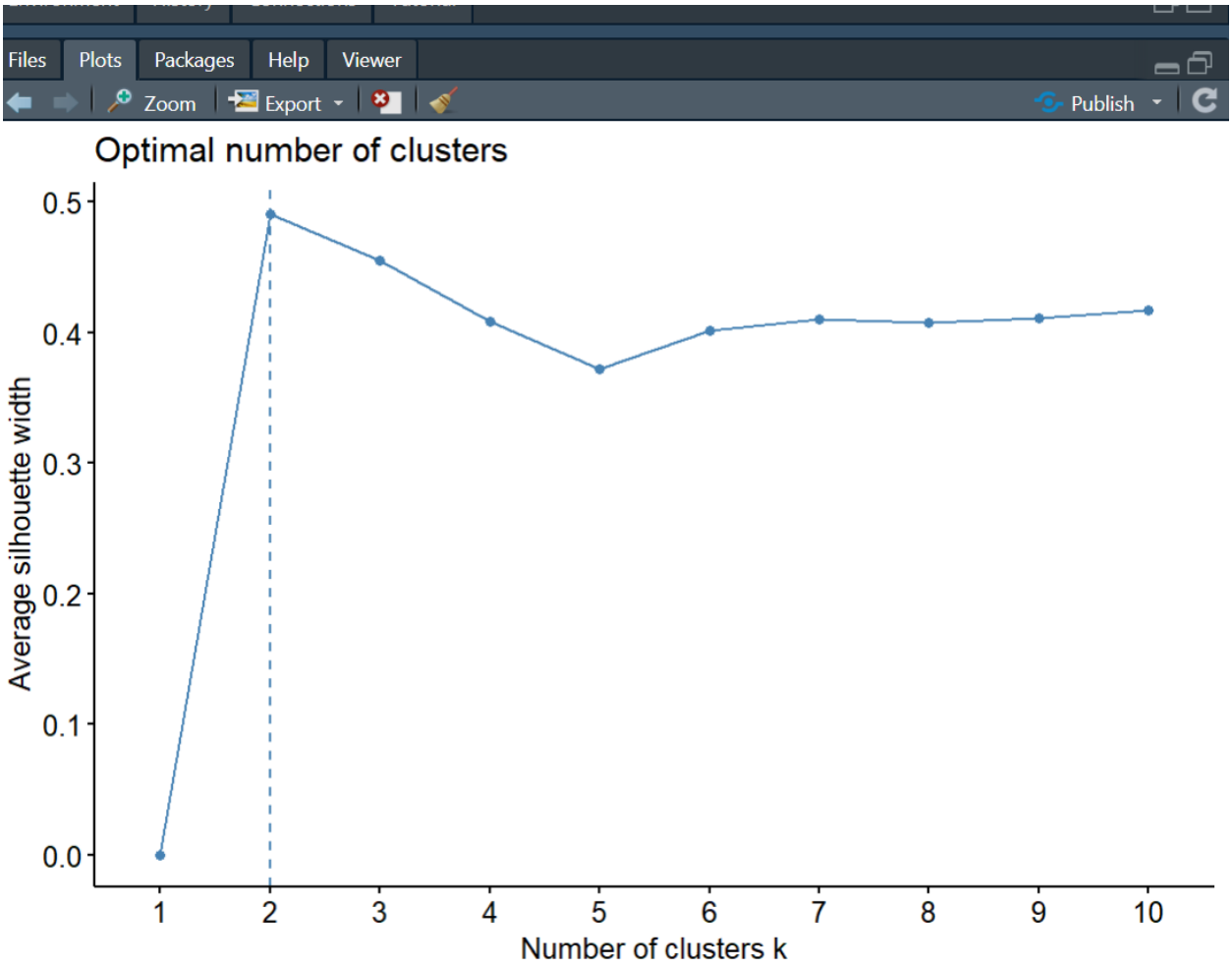
Silhouette Plot




Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Within groups sum of squares
0
100
200
300
400
5
10
15
20
Number of Clusters

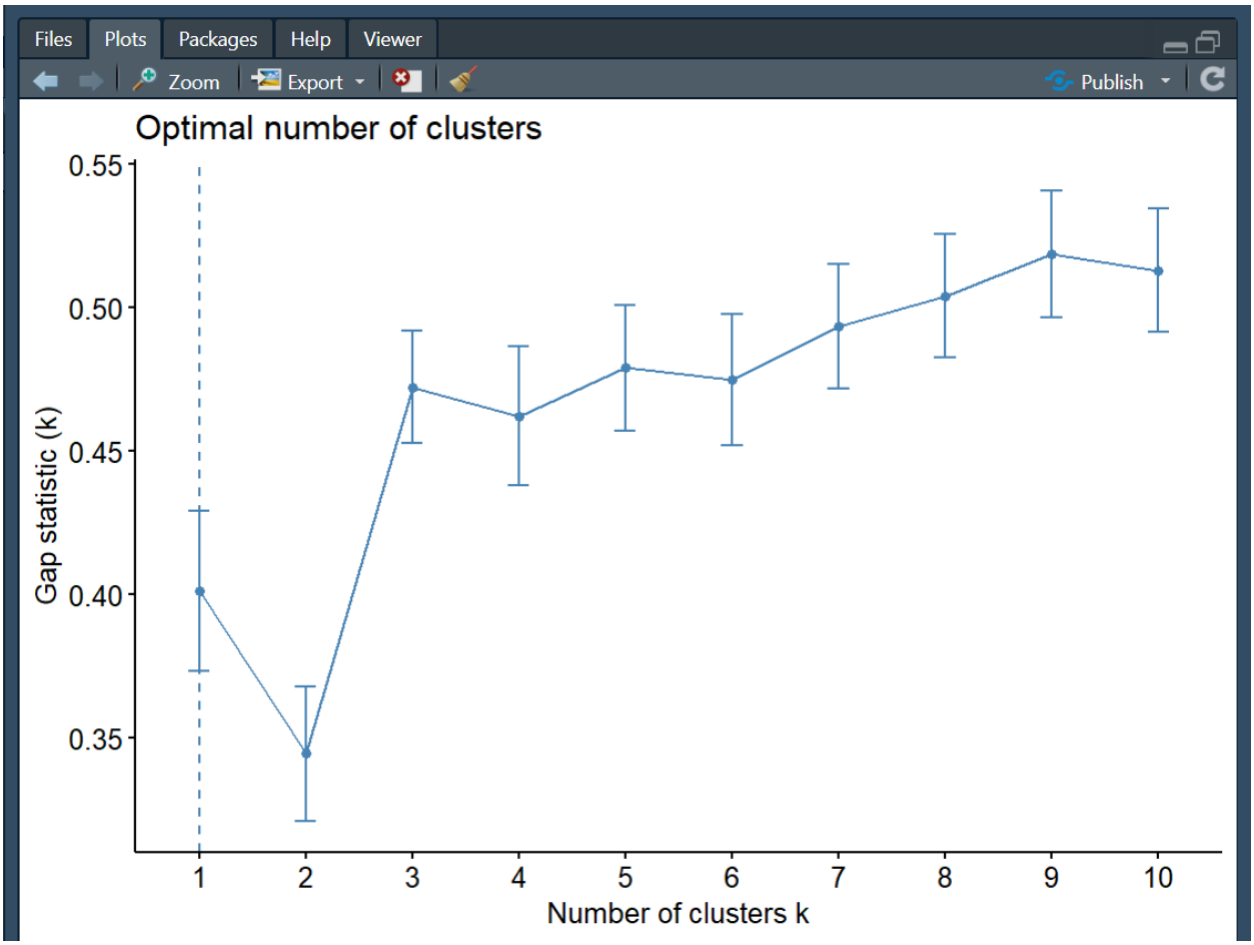

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Cluster Mapping
Component 2
-10
0
10
20
-40
-30
-20
-10
0
10
20
Component 1
These two components explain 91.46 % of the point variability.


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Optimal number of clusters
Total Within Sum of Square
400
300
200
100
1
2
3
4
5
6
7
8
9
10
Number of clusters k

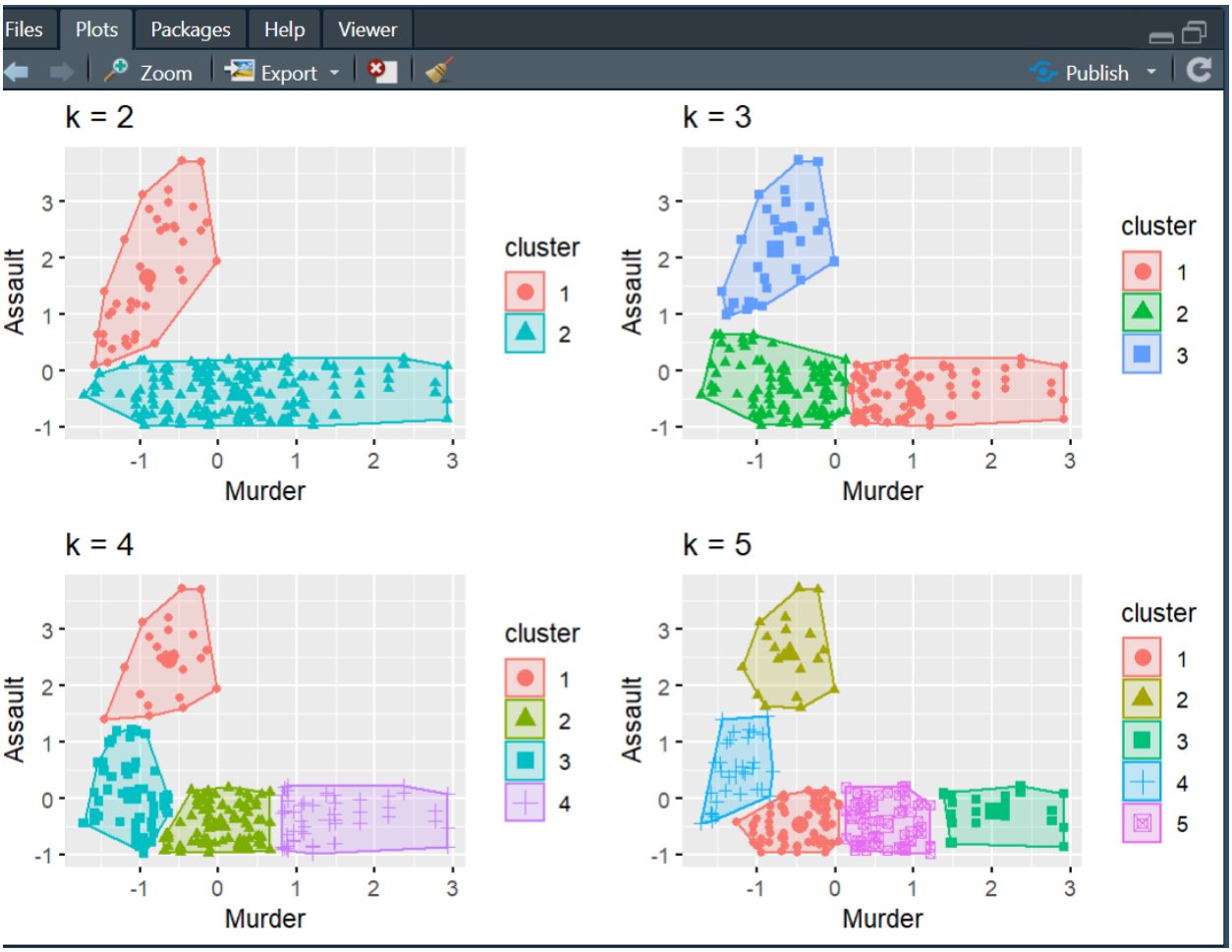


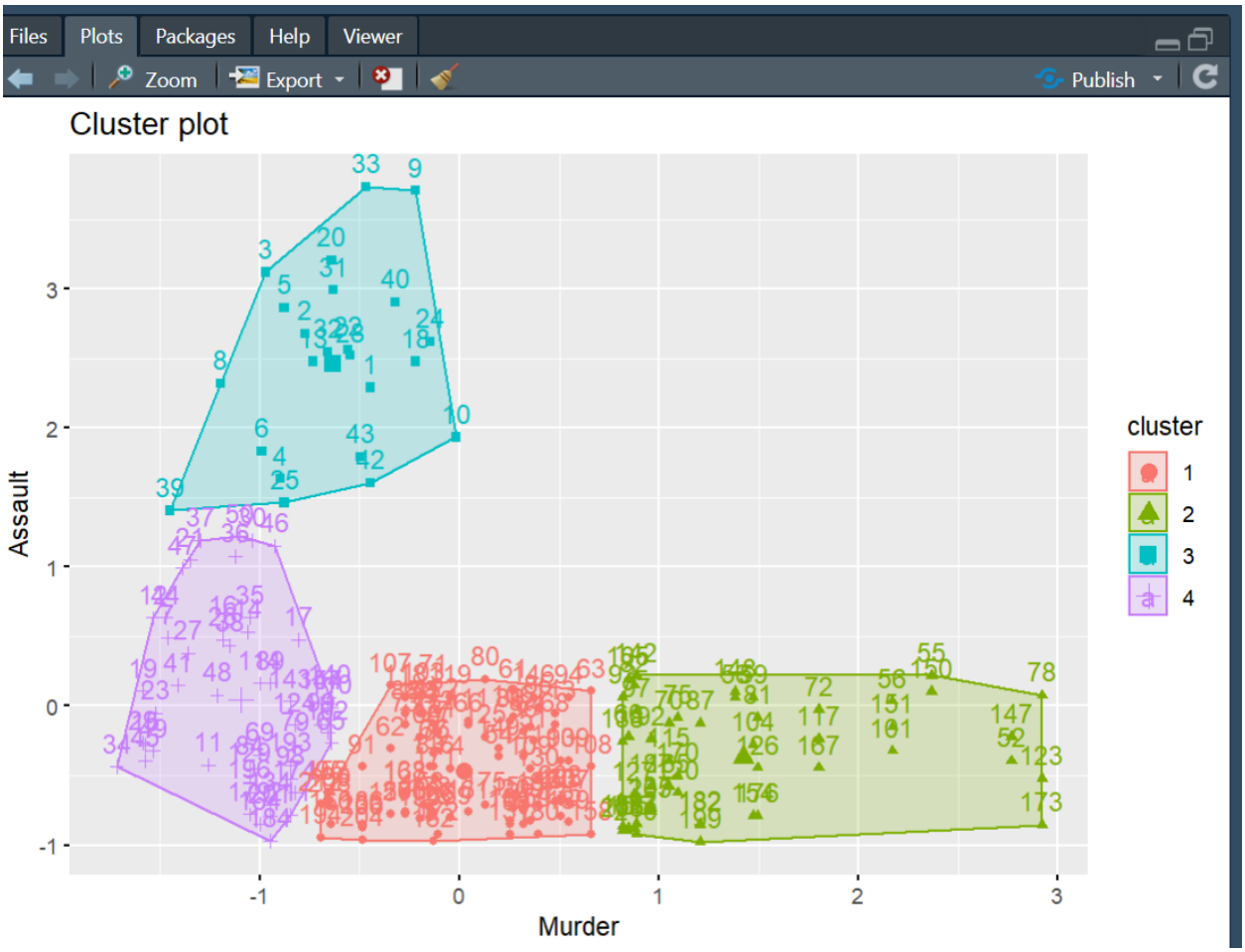
A high average silhouette width indicates a good clustering method that computes the average silhouette of observations for different values of k.

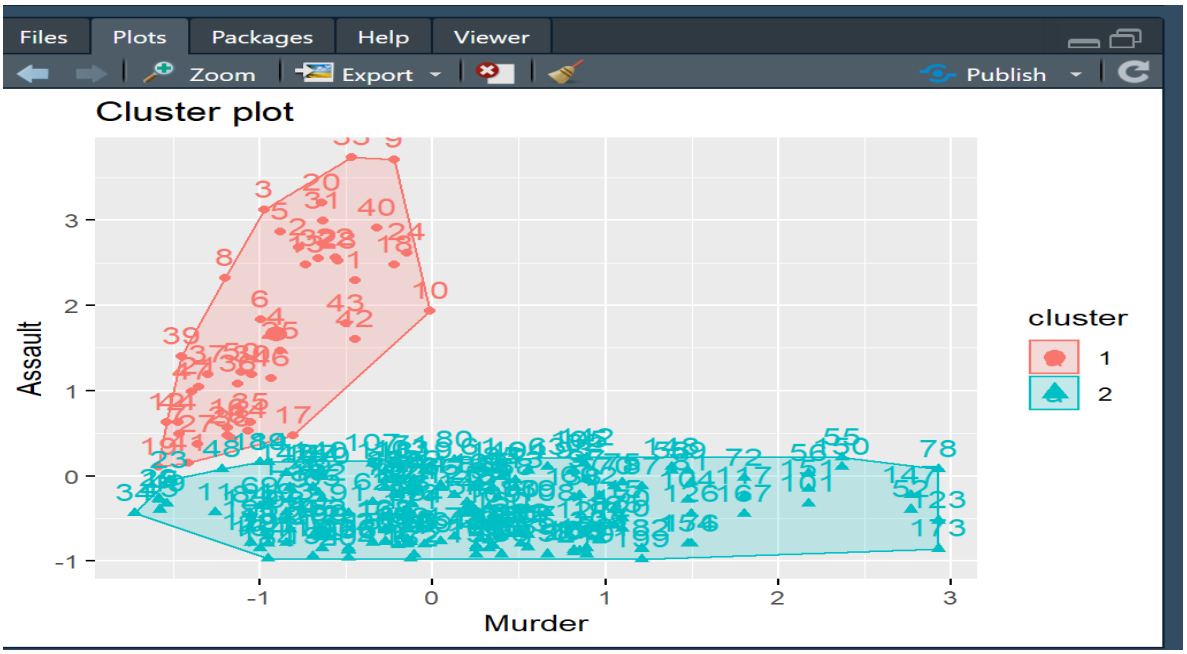
good clustering =2



In this method also optimal number of clusters is 2.


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
k = 2
k = 3
k = 4
k = 5
Assault
Murder
cluster
1
2
3
4
5


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Cluster plot
Assault
Murder
cluster
1
2
3
4


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Cluster plot
Assault
Murder
cluster
1
2

Results:

The good Clustering k=2

In this method also optimal number of clusters is 2. (Gap statics)

K=2

Cluster means:

```
     Murder    Assault
1 -0.9075537  1.6683283
2  0.2242191 -0.4121752


Clustering Gap statistic ["clusGap"] from call:
clusGap(x = nor, FUNcluster = kmeans, K.max = 10, B = 50, nstart = 25)
B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstmax'): 1
         logW   E.logW       gap     SE.sim
 [1,] 4.490161 4.891443 0.4012818 0.02791313
 [2,] 4.191047 4.535512 0.3444657 0.02349209
 [3,] 3.873402 4.345664 0.4722623 0.01965877
 [4,] 3.710750 4.172930 0.4621801 0.02433966
 [5,] 3.561039 4.039980 0.4789407 0.02189095
 [6,] 3.456945 3.931725 0.4747796 0.02298658
 [7,] 3.354487 3.847964 0.4934769 0.02189399
 [8,] 3.267765 3.771858 0.5040934 0.02144039
 [9,] 3.183247 3.701918 0.5186714 0.02201696
[10,] 3.124453 3.637372 0.5129192 0.02148391
```

Compute k-means clustering with k = 4

K-means clustering with 4 clusters of sizes 88, 49, 23, 52

Cluster means:

	Murder	Assault
1	0.02166337	-0.46487958
2	1.42295189	-0.36975467
3	-0.63486487	2.46482447
4	-1.09671397	0.04493114

References:

Violent Crime Rates by US State

2020 . English . Kaggle

Cluster Analysis in R

2021. English . Finnstats . R-bloggers