Clustering

Assignment 1

BP: Perform clustering for the crime data and identify the number of clusters formed and draw inferences.

PROCEDURE:

STEP 1: First we have to Exploratory Data Analysis which can be done by plotting scattered plot, box plots and summary.

summary(crime\_data\_1\_)

X1 Murder Assault UrbanPop

Length: 50 Min. : 0.800 Min. : 45.0 Min. :32.00

Class: character 1st Qu.: 4.075 1st Qu.:109.0 1st Qu.:54.50

Mode : character Median : 7.250 Median :159.0 Median :66.00

Mean : 7.788 Mean :170.8 Mean :65.54

3rd Qu.:11.250 3rd Qu.:249.0 3rd Qu.:77.75

Max. :17.400 Max. :337.0 Max. :91.00

Rape

Min. : 7.30

1st Qu.:15.07

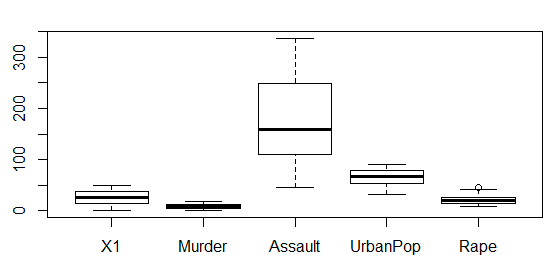
Median :20.10

Mean :21.23

3rd Qu.:26.18

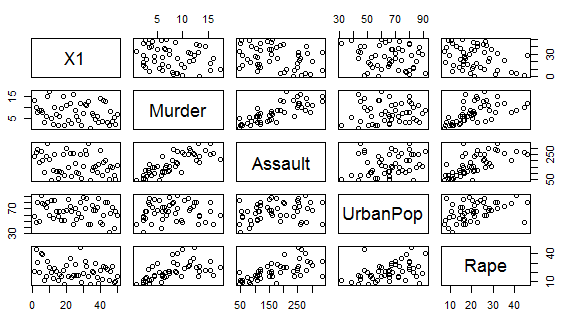
Max. :46.00

From the summary we can see that in assault the difference between mean and max is large so it may be right skewed it can my confirmed by using Box Plot .



It can be confirmed that Assault is right Skewed.

Scatter Plot for the Following data frame is:



STEP 2: Now we can observe from the summary that the data is not in similar scale so now we have to normalize the data into same scale which can be done as follows:

normzalizeddata <- scale(crime\_data\_1\_[,2:5])

View(normzalizeddata)

summary(normzalizeddata)

Murder Assault UrbanPop Rape

Min. :-1.6044 Min. :-1.5090 Min. :-2.31714 Min. :-1.4874

1st Qu.:-0.8525 1st Qu.:-0.7411 1st Qu.:-0.76271 1st Qu.:-0.6574

Median :-0.1235 Median :-0.1411 Median : 0.03178 Median :-0.1209

Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000

3rd Qu.: 0.7949 3rd Qu.: 0.9388 3rd Qu.: 0.84354 3rd Qu.: 0.5277

Max. : 2.2069 Max. : 1.9948 Max. : 1.75892 Max. : 2.6444

From the summary we can see that whole data is converted into Z Scale.

STEP 3: Now we have to measure the distance between the rows so that we can cluster so code is as follows:

d <- dist(normzalizeddata,method = "euclidean")

d

1 2 3 4 5 6 7 8

2 2.7037541

3 2.2935197 2.7006429

4 1.2898102 2.8260386 2.7177583

5 3.2631104 3.0125415 1.3104842 3.7636409

6 2.6510673 2.3265187 1.3650307 2.8310512 1.2876185

7 3.2152975 4.7399125 3.2628575 2.6076395 4.0663898 3.3279920

8 2.0192927 3.6213633 1.9093696 1.8003239 3.0737852 2.5547456 1.7568475

9 2.2981353 2.9967642 1.7493928 3.3721968 2.0250039 2.4458600 4.4700701 3.0614170

10 1.1314351 2.8194388 2.7871963 2.2117614 3.3780585 2.8649105 3.9738227 2.9838715

11 3.3885300 4.5301340 3.2621208 2.9723097 3.6589083 2.8233524 1.3843291 2.4748807

12 2.9146623 4.0580555 3.5210071 1.7687255 4.4879436 3.4767685 1.6354214 2.0382540

13 1.8734993 3.2670626 1.0825512 2.4626424 1.9117469 1.7898322 2.7400560 1.5584719

14 2.0761411 3.3655952 2.6407486 1.4450503 3.4061273 2.3655622 1.6147898 1.6973340

15 3.4878952 4.7251910 4.1157513 2.4252661 4.9708591 3.9406898 1.5470089 2.6068606

16 2.2941096 3.6808173 2.7762838 1.5718411 3.6071725 2.6272281 1.2280424 1.5510864

17 1.8475879 3.5440903 3.3567681 1.0598104 4.2463809 3.2274013 2.3346386 2.2514939

18 0.7722224 2.9631431 2.2178519 2.0254276 3.0176625 2.6546743 3.5329409 2.3266996

19 3.4851115 4.8322605 4.2961903 2.3621893 5.2699843 4.2713441 1.8792141 2.6560808

20 1.2896460 2.2777590 1.2117356 2.0582244 2.2312581 1.9667562 3.4968269 1.9624834

21 2.9874810 4.3729925 2.5162281 2.6881270 3.2156499 2.6522793 0.9468199 1.4382527

9 10 11 12 13 14 15 16

2

3

4

5

6

7

8

9

10 2.1812958

11 4.3596338 3.8105218

12 4.6999827 3.8005715 2.3658101

13 1.7711863 2.3135778 2.7329756 3.2728945

14 3.6150778 2.6924143 1.5460727 1.4923351 2.2027081

15 5.2682765 4.2517889 2.1564575 0.8584962 3.7380070 1.7786548

16 3.8424558 3.0071474 1.4648766 1.2103118 2.3228505 0.4287712 1.4699265

17 3.9474983 2.4408198 2.5203345 1.6565236 2.8478883 1.1790552 1.9426473 1.3020180

18 1.7529677 0.8592544 3.5687157 3.5283772 1.6535178 2.4957547 4.0359614 2.7284126

19 5.3946798 4.3334217 2.7160558 0.8486112 3.9342034 2.1029158 0.6457158 1.7913753

20 1.4355204 1.8388691 3.6148670 3.4014584 1.3429997 2.5430878 4.0642448 2.7400943

21 3.7753087 3.6706708 1.3276676 2.2201020 2.0080982 1.6615695 2.3510287 1.4343401

17 18 19 20 21 22 23 24

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18 2.4221964

19 1.9925855 4.0901924

20 2.8229479 1.2739137 4.1259083

21 2.6284451 3.1524549 2.6920282 2.9743193

25 26 27 28 29 30 31 32

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

33 34 35 36 37 38 39 40

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

41 42 43 44 45 46 47 48

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

49

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

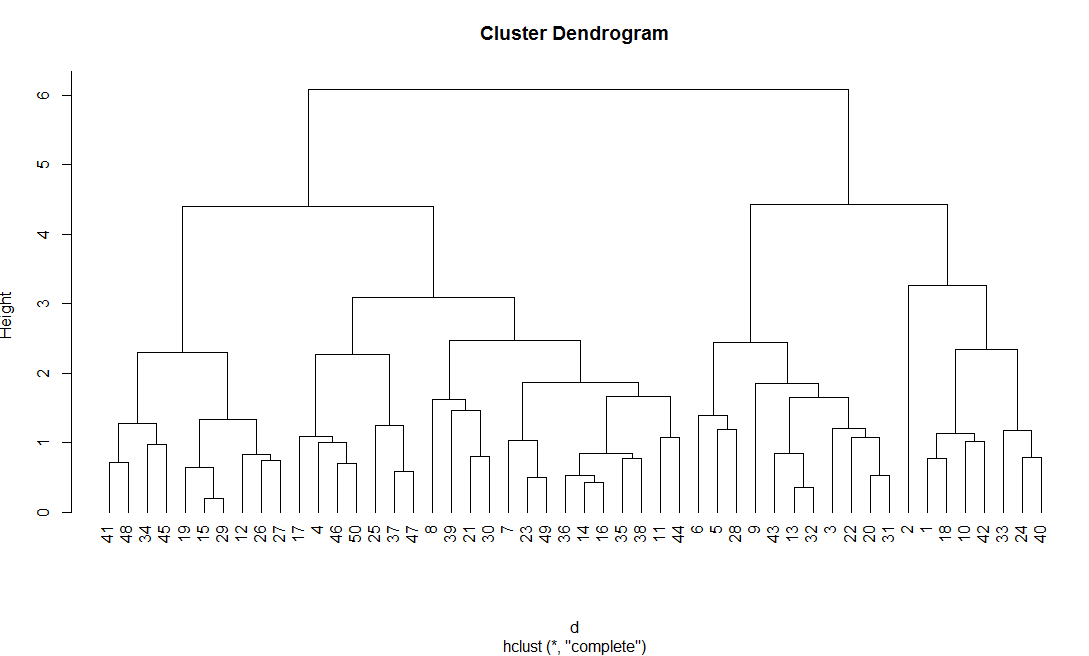
19

20

21

STEP 4: After the distance matrix is formed now we have to cluster it now:

|  |
| --- |
| cluster <- hclust(d,method = "complete")  > plot(cluster)  > plot(cluster,hang = -1) |
|  |
| |  | | --- | |  | |



STEP 5: Now after the Dendrogram is formed we can cut them into number of clusters which is suitable according to the data set :

groups <- cutree(cluster , k=10)

> groups

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

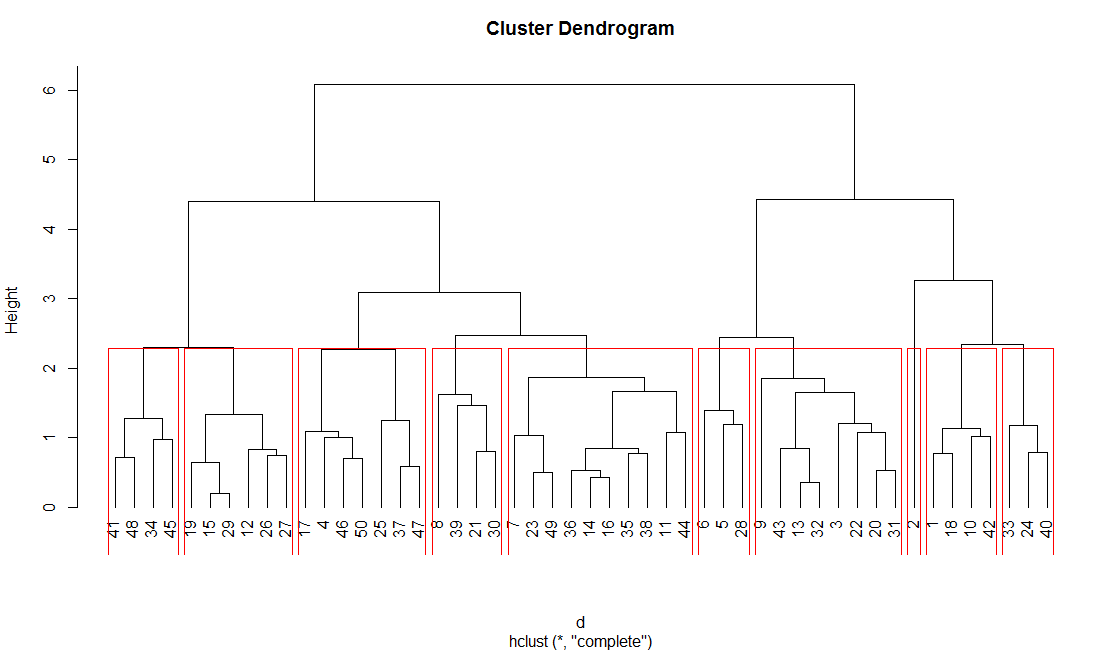
1 2 3 4 5 5 6 7 3 1 6 8 3 6 8 6 4 1 8 3 7 3 6 9 4 8 8 5

29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

8 7 3 3 9 10 6 6 4 6 7 9 10 1 3 6 10 4 4 10 6 4

Now there are 10 groups which are formed from the 50 entries of the data set.

To visualize the groups we use:

rect.hclust(cluster, k=10 , border = "red")

Now the groups are converted into column format :

clusters <- as.matrix(groups)

> clusters

[,1]

1 1

2 2

3 3

4 4

5 5

6 5

7 6

8 7

9 3

10 1

11 6

12 8

13 3

14 6

15 8

16 6

17 4

18 1

19 8

20 3

21 7

22 3

23 6

24 9

25 4

26 8

27 8

28 5

29 8

30 7

31 3

32 3

33 9

34 10

35 6

36 6

37 4

38 6

39 7

40 9

41 10

42 1

43 3

44 6

45 10

46 4

47 4

48 10

49 6

50 4

Now , for comparison of the final clusters we use the following code :

aggregate(crime\_data\_1\_[,2:5] , by=list(final$clusters) , FUN=mean)

Group Murder Assault UrbanPop Rape

1 1 14.800000 221.00000 60.75000 24.02500

2 2 10.000000 263.00000 48.00000 44.50000

3 3 11.562500 271.62500 77.50000 29.18750

4 4 7.385714 156.85714 62.14286 22.25714

5 5 9.700000 244.00000 83.33333 41.76667

6 6 5.050000 100.60000 71.80000 17.52000

7 7 5.275000 180.00000 83.25000 14.80000

8 8 3.216667 87.83333 55.50000 12.61667

9 9 14.500000 291.66667 45.66667 18.56667

10 10 3.125000 65.00000 40.00000 10.15000

Now we can see that clusters have been formed:

It can be inferred that group1 consists of max amount of murder % and group 10 consists of min murder % and so on .