Solution to COMP5121 Assignment 3

QING Pei, 11500811G

November 9, 2011

1 Part A (Individual)

In any later distance calculations, the data should be normalized. The Mahalanobis distance is used here, which is defined as:

$$d(\vec{x}, \vec{y}) = \sqrt{\Sigma_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}} \tag{1}$$

σ_i^2 is the variance of the i^{th} item in the vector.
First, calculate the variance of each column in the data table. The last row in

Table 1: Original data

No.	b	c	d	e	f	g
1	16.9	4.36	2.73	155	350	8
2	15.5	4.054	2.26	142	351	8
3	18.5	3.94	2.45	150	360	8
4	30	2.155	3.7	68	98	4
5	30.9	2.23	3.37	75	105	4
6	20.6	3.38	2.73	105	231	6
7	20.8	3.07	3.08	85	200	6
8	18.1	3.41	2.73	120	258	6
9	16.5	3.955	2.26	138	351	8
10	35.1	1.915	2.97	80	98	4
11	27.4	2.67	3.08	80	121	4
12	29.5	2.135	3.05	68	98	4
13	28.4	2.67	2.53	90	151	4
14	26.8	2.7	2.84	115	173	6
15	34.2	2.2	3.37	70	105	4
variance	42.666	0.617	0.161	931.262	10432.889	2.773

Table 1 is variance of the corresponding column.
The k-means algorithm is described as:

Make an initial guess of the partition.
while Any one of the centroids moves from its last position **do**
  Place each point in the space represented by the closest centroid;
  Update centroid position as the mean of all points in its space.
end while

1.1 Manual k-means, k=2

Table 2: Cluster mean @ step 1

mean1	16.9	4.36	2.73	155	350	8
mean2	15.5	4.054	2.26	142	351	8

Table 3: Clustering result @ step 1

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	**cluster@step**
1	0.000	1.752	1
2	1.752	0.000	2
3	0.869	0.533	2
4	37.733	41.440	1
5	32.889	34.996	1
6	7.362	7.011	2
7	12.676	13.522	1
8	5.066	4.994	2
9	1.952	0.057	2
10	35.709	35.585	2
11	24.811	25.569	1
12	32.367	32.226	2
13	22.080	19.965	2
14	13.003	13.317	1
15	36.404	38.559	1

Table 4: Cluster mean @ step 2

mean1	26.714	2.769	3.167	92.571	164.571	5.143
mean2	22.775	3.182	2.623	111.625	237.250	6.000

Take the first two points as initial centroids shown in Table 2.
Calculate the distance square (a sqrt() is saved here since we only need to know which is closer but not the actual distance) to means, shown in Table 3 and assign each point to a cluster.
The temporary partition at each step is shown in Table 5,7 and 9. The updated centroid positions are shown in Table 4,6 and 8.
The centroid stay unchanged in the last step. The clustering result is shown in Table 10.

1.2 Manual k-means, k=3

Take the last three points as initial centroids shown in Table 11:
Calculate the distance square to means, shown in Table 12 and assign each point to a cluster.
The temporary partition at each step is shown in Table 14,16,18,20 and 22. The updated centroid positions are shown in Table 13,15,17,19 and 21.

Table 5: Clustering result @ step 2

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	**cluster@step**
1	17.971	7.810	2
2	19.634	6.962	2
3	17.144	6.012	2
4	4.173	15.491	1
5	2.280	11.048	1
6	3.522	0.297	2
7	1.460	2.306	1
8	5.501	0.785	2
9	18.327	6.136	2
10	4.138	11.289	1
11	0.897	6.039	1
12	2.463	9.318	1
13	3.100	3.878	1
14	1.485	1.459	2
15	3.453	13.074	1

Table 6: Cluster mean @ step 3

mean1	29.538	2.381	3.144	77.000	122.000	4.250
mean2	18.986	3.686	2.571	132.143	296.286	7.143

Table 7: Clustering result @ step 3

No.	$dist^2$ to mean1	$dist^2$ to mean 2	cluster@step
1	27.745	2.098	2
2	28.644	1.764	2
3	26.010	1.199	2
4	2.174	26.301	1
5	0.453	21.298	1
6	7.640	2.039	2
7	4.341	6.044	1
8	10.710	1.067	2
9	26.947	1.454	2
10	1.352	22.405	1
11	0.300	14.364	1
12	0.317	19.660	1
13	2.791	11.252	1
14	3.818	5.697	1
15	0.983	24.180	1

Table 8: Cluster mean @ step 4

mean1	29.233	2.416	3.110	81.222	127.667	4.444
mean2	17.683	3.850	2.527	135.000	316.833	7.333

Table 9: Clustering result @ step 4

No.	$dist^2$ to mean1	$dist^2$ to mean 2	cluster@step
1	25.730	1.388	2
2	26.564	0.946	2
3	23.983	0.646	2
4	2.630	30.181	1
5	0.703	24.939	1
6	6.653	3.128	2
7	3.755	7.749	1
8	9.519	1.789	2
9	24.929	0.774	2
10	1.493	26.245	1
11	0.266	17.302	1
12	0.495	23.158	1
13	2.417	13.765	1
14	3.017	7.755	1
15	1.329	28.069	1

Table 10: Clustering result of manual k-means, k=2

cluster1	4	5	7	10	11	12	13	14	15
cluster2	1	2	3	6	8	9			

Table 11: Cluster mean @ step 1

mean1	28.400	2.670	2.530	90.000	151.000	4.000
mean2	26.800	2.700	2.840	115.000	173.000	6.000
mean3	34.200	2.200	3.370	70.000	105.000	4.000

Table 12: Clustering result @ step 1

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	22.080	13.003	36.404	2
2	19.965	13.317	38.559	2
3	18.774	11.162	34.818	2
4	9.782	9.669	1.102	3
5	5.288	6.101	0.284	3
6	4.789	2.156	13.416	2
7	5.192	2.460	8.507	2
8	7.129	3.386	17.364	2
9	18.526	12.177	36.525	2
10	3.556	6.016	1.257	3
11	2.096	3.385	2.096	1
12	2.961	5.316	1.170	3
13	0.000	2.818	6.162	1
14	2.818	0.000	7.494	2
15	6.162	7.494	0.000	3

Table 13: Cluster mean @ step 2

mean1	27.900	2.670	2.805	85.000	136.000	4.000
mean2	19.213	3.609	2.635	126.250	284.250	7.000
mean3	31.940	2.127	3.292	72.200	100.800	4.000

Table 14: Clustering result @ step 2

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	22.922	2.759	34.431	2
2	22.243	2.572	35.972	2
3	20.584	1.919	32.675	2
4	5.958	23.413	1.143	3
5	2.707	18.784	0.090	3
6	4.838	1.303	11.744	2
7	3.746	4.628	7.191	1
8	7.358	0.618	15.385	2
9	20.784	2.176	34.039	2
10	2.474	20.131	1.017	3
11	0.524	12.326	1.345	1
12	1.346	17.285	0.523	3
13	0.524	9.833	4.960	1
14	2.577	4.632	6.330	1
15	3.605	21.560	0.173	3

Table 15: Cluster mean @ step 3

mean1	25.850	2.778	2.883	92.500	161.250	5.000
mean2	17.683	3.850	2.527	135.000	316.833	7.333
mean3	31.940	2.127	3.292	72.200	100.800	4.000

Table 16: Clustering result @ step 3

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	16.937	1.388	34.431	2
2	16.887	0.946	35.972	2
3	15.201	0.646	32.675	2
4	6.572	30.181	1.143	3
5	3.553	24.939	0.090	3
6	2.374	3.128	11.744	1
7	1.544	7.749	7.191	1
8	4.271	1.789	15.385	2
9	15.623	0.774	34.039	2
10	4.171	26.245	1.017	3
11	1.001	17.302	1.345	1
12	2.544	23.158	0.523	3
13	1.320	13.765	4.960	1
14	0.960	7.755	6.330	1
15	4.859	28.069	0.173	3

Table 17: Cluster mean @ step 4

mean1	24.800	2.898	2.852	95.000	175.200	5.200
mean2	17.100	3.944	2.486	141.000	334.000	7.600
mean3	31.940	2.127	3.292	72.200	100.800	4.000

Table 18: Clustering result @ step 4

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	14.642	0.944	34.431	2
2	14.532	0.483	35.972	2
3	13.043	0.263	32.675	2
4	7.869	33.977	1.143	3
5	4.683	28.457	0.090	3
6	1.519	4.504	11.744	1
7	1.143	9.762	7.191	1
8	3.129	2.805	15.385	2
9	13.378	0.421	34.039	2
10	5.472	29.729	1.017	3
11	1.608	20.326	1.345	3
12	3.578	26.618	0.523	3
13	1.634	16.311	4.960	1
14	0.819	9.625	6.330	1
15	6.190	31.750	0.173	3

Table 19: Cluster mean @ step 5

mean1	24.150	2.955	2.795	98.750	188.750	5.500
mean2	17.100	3.944	2.486	141.000	334.000	7.600
mean3	31.183	2.218	3.257	73.500	104.167	4.000

Table 20: Clustering result @ step 5

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	12.602	0.944	32.641	2
2	12.275	0.483	34.051	2
3	10.946	0.263	30.950	2
4	9.543	33.977	1.296	3
5	6.063	28.457	0.084	3
6	0.918	4.504	10.589	1
7	1.094	9.762	6.364	1
8	2.254	2.805	14.073	1
9	11.202	0.421	32.195	2
10	6.732	29.729	1.067	3
11	2.513	20.326	0.934	3
12	4.781	26.618	0.379	3
13	2.021	16.311	4.296	1
14	0.680	9.625	5.652	1
15	7.716	31.750	0.307	3

Table 21: Cluster mean @ step 6

mean1	22.940	3.046	2.782	103.000	202.600	5.600
mean2	16.850	4.077	2.425	146.250	353.000	8.000
mean3	31.183	2.218	3.257	73.500	104.167	4.000

Table 22: Clustering result @ step 6

No.	$dist^2$ **to mean1**	$dist^2$ **to mean 2**	$dist^2$ **to mean 3**	**cluster@step**
1	10.734	0.791	32.641	2
2	10.458	0.232	34.051	2
3	9.266	0.118	30.950	2
4	10.977	38.719	1.296	3
5	7.390	32.822	0.084	3
6	0.465	6.392	10.589	1
7	1.066	12.390	6.364	1
8	1.443	4.384	14.073	1
9	9.508	0.270	32.195	2
10	8.299	33.946	1.067	3
11	3.376	24.126	0.934	3
12	6.087	30.870	0.379	3
13	2.682	19.484	4.296	1
14	0.860	12.062	5.652	1
15	9.285	36.224	0.307	3

Table 23: Clustering result of manual k-means, k=3

cluster1	6	7	8	13	14	
cluster2	1	2	3	9		
cluster3	4	5	10	11	12	15

The centroid stay unchanged in the last step. The clustering result is shown in Table 23.

1.3 Software k-means, k=2&3

Read the date file:

```
> cars <- read.csv("question1_data.csv")
> cars
```

```
      b     c    d   e   f g
1  16.9 4.360 2.73 155 350 8
2  15.5 4.054 2.26 142 351 8
3  18.5 3.940 2.45 150 360 8
4  30.0 2.155 3.70  68  98 4
5  30.9 2.230 3.37  75 105 4
6  20.6 3.380 2.73 105 231 6
7  20.8 3.070 3.08  85 200 6
8  18.1 3.410 2.73 120 258 6
9  16.5 3.955 2.26 138 351 8
10 35.1 1.915 2.97  80  98 4
11 27.4 2.670 3.08  80 121 4
12 29.5 2.135 3.05  68  98 4
13 28.4 2.670 2.53  90 151 4
14 26.8 2.700 2.84 115 173 6
15 34.2 2.200 3.37  70 105 4
```

Normalize the figures:

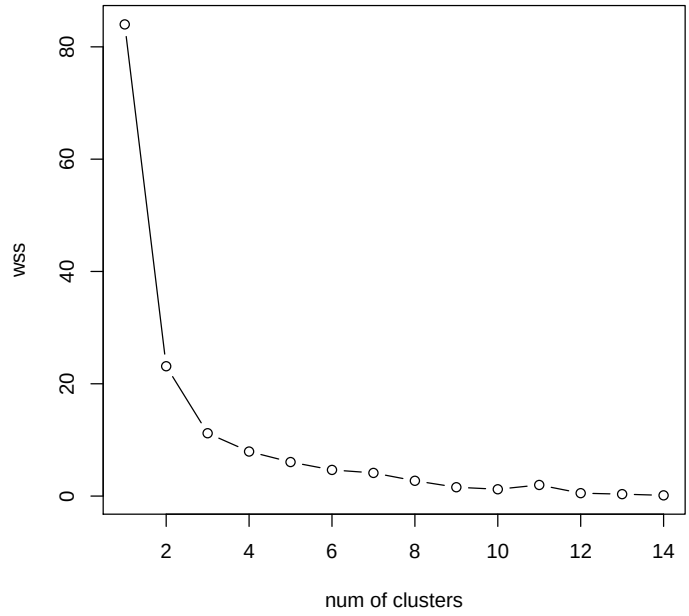
```
> options(digits = 2)
> cars.scale <- scale(cars)
> cars.scale
```

```
          b      c      d      e      f     g
 [1,] -1.14  1.686 -0.353  1.655  1.387  1.39
 [2,] -1.35  1.309 -1.485  1.243  1.397  1.39
 [3,] -0.90  1.169 -1.027  1.496  1.482  1.39
 [4,]  0.80 -1.027  1.982 -1.100 -0.996 -0.93
 [5,]  0.93 -0.934  1.188 -0.878 -0.930 -0.93
 [6,] -0.59  0.480 -0.353  0.072  0.262  0.23
 [7,] -0.56  0.099  0.490 -0.561 -0.032  0.23
 [8,] -0.96  0.517 -0.353  0.547  0.517  0.23
 [9,] -1.20  1.188 -1.485  1.116  1.397  1.39
[10,]  1.55 -1.322  0.225 -0.720 -0.996 -0.93
[11,]  0.41 -0.393  0.490 -0.720 -0.779 -0.93
[12,]  0.72 -1.051  0.417 -1.100 -0.996 -0.93
[13,]  0.56 -0.393 -0.835 -0.403 -0.495 -0.93
[14,]  0.32 -0.356 -0.088  0.388 -0.287  0.23
[15,]  1.42 -0.971  1.188 -1.036 -0.930 -0.93
attr(,"scaled:center")
    b     c     d     e     f     g
```

```
 24.6   3.0   2.9 102.7 203.3   5.6
attr(,"scaled:scale")
     b      c      d      e      f      g
  6.76   0.81   0.42  31.59 105.73   1.72
```

How within-group sum of squares change with the number of clusters:

```
> wss <- (nrow(cars.scale) - 1) * sum(apply(cars.scale, 2, var))
> for (i in 1:14) wss[i] <- sum(kmeans(cars.scale, centers = i)$withinss)
> plot(1:14, wss, type = "b", xlab = "num of clusters", ylab = "wss")
```



K-means, k=2:

```
> fit2 <- kmeans(cars.scale, 2)
> aggregate(cars.scale, by = list(fit2$cluster), FUN = mean)

  Group.1     b     c     d     e     f     g
1       1 -1.02  1.06 -0.84  1.02  1.07  1.01
2       2  0.68 -0.71  0.56 -0.68 -0.72 -0.67

> two_cluster <- data.frame(cars.scale, fit2$cluster)
> two_cluster

       b      c      d      e      f     g fit2.cluster
1  -1.14  1.686 -0.353  1.655  1.387  1.39           1
2  -1.35  1.309 -1.485  1.243  1.397  1.39           1
3  -0.90  1.169 -1.027  1.496  1.482  1.39           1
4   0.80 -1.027  1.982 -1.100 -0.996 -0.93           2
5   0.93 -0.934  1.188 -0.878 -0.930 -0.93           2
```

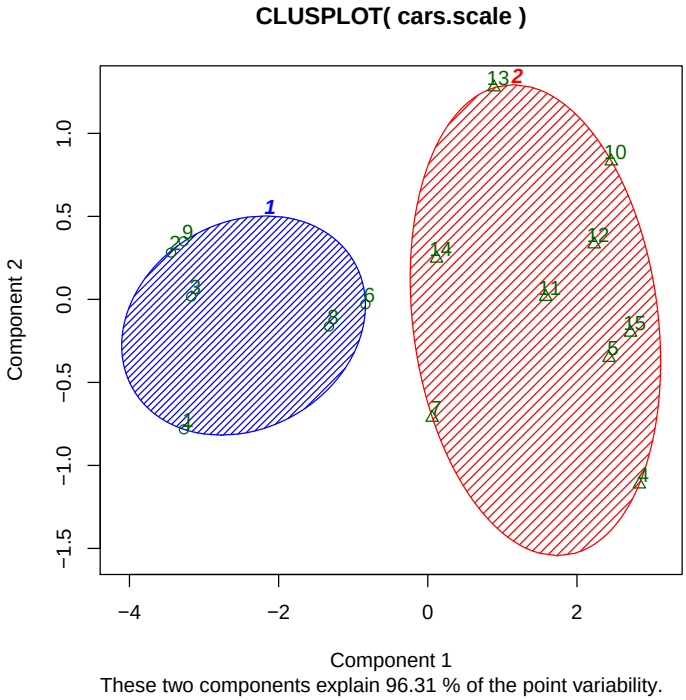
```
6  -0.59  0.480 -0.353  0.072  0.262  0.23           1
7  -0.56  0.099  0.490 -0.561 -0.032  0.23           2
8  -0.96  0.517 -0.353  0.547  0.517  0.23           1
9  -1.20  1.188 -1.485  1.116  1.397  1.39           1
10  1.55 -1.322  0.225 -0.720 -0.996 -0.93           2
11  0.41 -0.393  0.490 -0.720 -0.779 -0.93           2
12  0.72 -1.051  0.417 -1.100 -0.996 -0.93           2
13  0.56 -0.393 -0.835 -0.403 -0.495 -0.93           2
14  0.32 -0.356 -0.088  0.388 -0.287  0.23           2
15  1.42 -0.971  1.188 -1.036 -0.930 -0.93           2
```

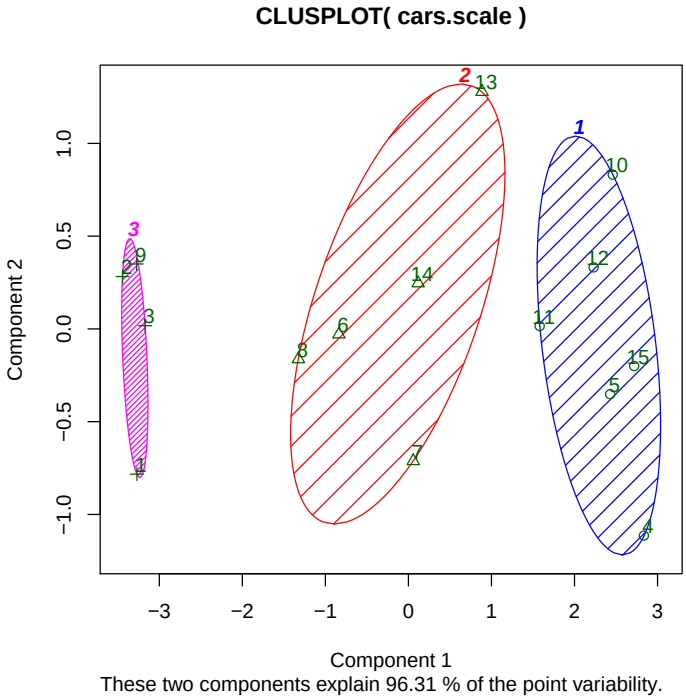
K-means, k=3:

```
> fit3 <- kmeans(cars.scale, 3)
> aggregate(cars.scale, by = list(fit3$cluster), FUN = mean)

  Group.1     b      c     d       e       f        g
1       1  0.97 -0.950  0.91 -0.9255 -0.9380 -9.3e-01
2       2 -0.25  0.069 -0.23  0.0084 -0.0069  2.0e-16
3       3 -1.15  1.338 -1.09  1.3776  1.4156  1.4e+00

> three_cluster <- data.frame(cars.scale, fit3$cluster)
> three_cluster

       b      c      d      e      f     g fit3.cluster
1  -1.14  1.686 -0.353  1.655  1.387  1.39           3
2  -1.35  1.309 -1.485  1.243  1.397  1.39           3
3  -0.90  1.169 -1.027  1.496  1.482  1.39           3
4   0.80 -1.027  1.982 -1.100 -0.996 -0.93           1
5   0.93 -0.934  1.188 -0.878 -0.930 -0.93           1
6  -0.59  0.480 -0.353  0.072  0.262  0.23           2
7  -0.56  0.099  0.490 -0.561 -0.032  0.23           2
8  -0.96  0.517 -0.353  0.547  0.517  0.23           2
9  -1.20  1.188 -1.485  1.116  1.397  1.39           3
10  1.55 -1.322  0.225 -0.720 -0.996 -0.93           1
11  0.41 -0.393  0.490 -0.720 -0.779 -0.93           1
12  0.72 -1.051  0.417 -1.100 -0.996 -0.93           1
13  0.56 -0.393 -0.835 -0.403 -0.495 -0.93           2
14  0.32 -0.356 -0.088  0.388 -0.287  0.23           2
15  1.42 -0.971  1.188 -1.036 -0.930 -0.93           1
```

1.4 Review of k-means results

First thing to note is that manual and software k-means give the same clustering results, which proves k-means is relatively stable.
Visualize the 2-cluster and 3-cluster partition, we can see the difference.





To compare the two clustering results, we need to check some attributes:

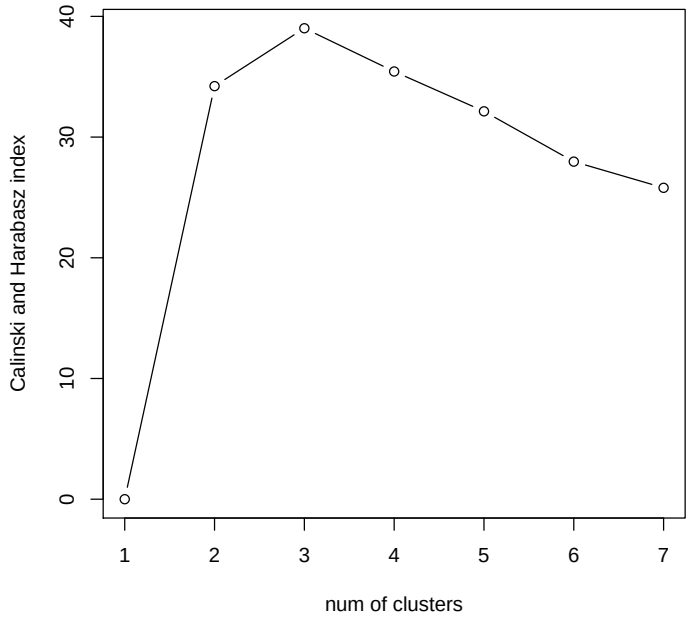
```
> fit2$withinss/fit2$betweenss
```

```
[1] 0.13 0.25

> fit3$withinss/fit3$betweenss

[1] 0.052 0.084 0.018
```

The within/between distance ratio shows that the 3-cluster result is better. However, there are some better criteria for comparison. E.g. the Calinski and Harabasz index.
B(k) = between cluster sum of squares
W(k) = within cluster sum of squares
Maximize CH(k) over the clusters:

$$CH(k) = \frac{B(k)/(k-1)}{W(k)/(n-k)} \tag{2}$$

With the help of GNU/R, we can obtain different measures of attributes of the partitions. Try increase the number of clusters and we can see the Calinski and Harabasz index reaches maximum at k=3.



Therefore, for this dataset, k=3 is an optimal parameter.

1.5 Hierarchical Agglomerative Single-Linkage

The initial distance matrix is shown in Table 24.
2 and 9 are closest to each other, so assign them to the same cluster. Now the partition is 2,9 and the rest are individual points. Update the distance matrix and it becomes Table 25.

Table 24: Distance matrix @ initial

d_mat	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.00														
2	1.32	0.00													
3	0.93	0.73	0.00												
4	6.14	6.44	6.13	0.00											
5	5.73	5.92	5.63	0.87	0.00										
6	2.71	2.65	2.51	3.87	3.33	0.00									
7	3.56	3.68	3.50	2.91	2.54	1.20	0.00								
8	2.25	2.23	2.09	4.30	3.81	0.68	1.66	0.00							
9	1.40	0.24	0.69	6.29	5.76	2.51	3.54	2.13	0.00						
10	5.98	5.97	5.69	2.04	1.26	3.55	3.08	4.05	5.79	0.00					
11	4.98	5.06	4.84	1.78	1.08	2.44	1.83	2.94	4.91	1.56	0.00				
12	5.69	5.68	5.47	1.62	0.87	3.10	2.44	3.62	5.51	1.00	0.88	0.00			
13	4.70	4.47	4.33	3.13	2.30	2.19	2.28	2.67	4.30	1.89	1.45	1.72	0.00		
14	3.61	3.65	3.34	3.11	2.47	1.47	1.57	1.84	3.49	2.45	1.84	2.31	1.68	0.00	
15	6.03	6.21	5.90	1.05	0.53	3.66	2.92	4.17	6.04	1.12	1.45	1.08	2.48	2.74	0.00

Table 25: Distance matrix @ step 1

d_mat	1	2,9	3	4	5	6	7	8	10	11	12	13	14	15
1	0.00													
2,9	1.32	0.00												
3	0.93	0.69	0.00											
4	6.14	6.29	6.13	0.00										
5	5.73	5.76	5.63	0.87	0.00									
6	2.71	2.51	2.51	3.87	3.33	0.00								
7	3.56	3.54	3.50	2.91	2.54	1.20	0.00							
8	2.25	2.13	2.09	4.30	3.81	0.68	1.66	0.00						
10	5.98	5.79	5.69	2.04	1.26	3.55	3.08	4.05	0.00					
11	4.98	4.91	4.84	1.78	1.08	2.44	1.83	2.94	1.56	0.00				
12	5.69	5.51	5.47	1.62	0.87	3.10	2.44	3.62	1.00	0.88	0.00			
13	4.70	4.30	4.33	3.13	2.30	2.19	2.28	2.67	1.89	1.45	1.72	0.00		
14	3.61	3.49	3.34	3.11	2.47	1.47	1.57	1.84	2.45	1.84	2.31	1.68	0.00	
15	6.03	6.04	5.90	1.05	0.53	3.66	2.92	4.17	1.12	1.45	1.08	2.48	2.74	0.00

5 and 15 are closest to each other, so assign them to the same cluster. Now the partition is 2,9, 5,15 and the rest are individual points. Update the distance matrix and it becomes Table 26.

Table 26: Distance matrix @ step 2

d_mat	1	2,9	3	4	5,15	6	7	8	10	11	12	13	14
1	0.00												
2,9	1.32	0.00											
3	0.93	0.69	0.00										
4	6.14	6.29	6.13	0.00									
5,15	5.73	5.76	5.63	0.87	0.00								
6	2.71	2.51	2.51	3.87	3.33	0.00							
7	3.56	3.54	3.50	2.91	2.54	1.20	0.00						
8	2.25	2.13	2.09	4.30	3.81	0.68	1.66	0.00					
10	5.98	5.79	5.69	2.04	1.12	3.55	3.08	4.05	0.00				
11	4.98	4.91	4.84	1.78	1.08	2.44	1.83	2.94	1.56	0.00			
12	5.69	5.51	5.47	1.62	0.87	3.10	2.44	3.62	1.00	0.88	0.00		
13	4.70	4.30	4.33	3.13	2.30	2.19	2.28	2.67	1.89	1.45	1.72	0.00	
14	3.61	3.49	3.34	3.11	2.47	1.47	1.57	1.84	2.45	1.84	2.31	1.68	0.00

6 and 8 are closest to each other, so assign them to the same cluster. Now the partition is 2,9, 5,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 27.

Table 27: Distance matrix @ step 3

d_mat	1	2,9	3	4	5,15	6,8	7	10	11	12	13	14
1	0.00											
2,9	1.32	0.00										
3	0.93	0.69	0.00									
4	6.14	6.29	6.13	0.00								
5,15	5.73	5.76	5.63	0.87	0.00							
6,8	2.25	2.13	2.09	3.87	3.33	0.00						
7	3.56	3.54	3.50	2.91	2.54	1.20	0.00					
10	5.98	5.79	5.69	2.04	1.12	3.55	3.08	0.00				
11	4.98	4.91	4.84	1.78	1.08	2.44	1.83	1.56	0.00			
12	5.69	5.51	5.47	1.62	0.87	3.10	2.44	1.00	0.88	0.00		
13	4.70	4.30	4.33	3.13	2.30	2.19	2.28	1.89	1.45	1.72	0.00	
14	3.61	3.49	3.34	3.11	2.47	1.47	1.57	2.45	1.84	2.31	1.68	0.00

2,9 and 3 are closest to each other, so assign them to the same cluster. Now the partition is 2,3,9, 5,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 28.
5,15 and 12 are closest to each other, so assign them to the same cluster. Now the partition is 2,3,9, 5,12,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 29.
5,12,15 and 4 are closest to each other, so assign them to the same cluster. Now the partition is 2,3,9, 4,5,12,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 30.
4,5,12,15 and 11 are closest to each other, so assign them to the same cluster. Now the partition is 2,3,9, 4,5,11,12,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 31.

Table 28: Distance matrix @ step 4

d_mat	1	2,9,3	4	5,15	6,8	7	10	11	12	13	14
1	0.00										
2,9,3	0.93	0.00									
4	6.14	6.13	0.00								
5,15	5.73	5.63	0.87	0.00							
6,8	2.25	2.09	3.87	3.33	0.00						
7	3.56	3.50	2.91	2.54	1.20	0.00					
10	5.98	5.69	2.04	1.12	3.55	3.08	0.00				
11	4.98	4.84	1.78	1.08	2.44	1.83	1.56	0.00			
12	5.69	5.47	1.62	0.87	3.10	2.44	1.00	0.88	0.00		
13	4.70	4.30	3.13	2.30	2.19	2.28	1.89	1.45	1.72	0.00	
14	3.61	3.34	3.11	2.47	1.47	1.57	2.45	1.84	2.31	1.68	0.00

Table 29: Distance matrix @ step 5

d_mat	1	2,9,3	4	5,15,12	6,8	7	10	11	13	14
1	0.00									
2,9,3	0.93	0.00								
4	6.14	6.13	0.00							
5,15,12	5.69	5.47	0.87	0.00						
6,8	2.25	2.09	3.87	3.10	0.00					
7	3.56	3.50	2.91	2.44	1.20	0.00				
10	5.98	5.69	2.04	1.00	3.55	3.08	0.00			
11	4.98	4.84	1.78	0.88	2.44	1.83	1.56	0.00		
13	4.70	4.30	3.13	1.72	2.19	2.28	1.89	1.45	0.00	
14	3.61	3.34	3.11	2.31	1.47	1.57	2.45	1.84	1.68	0.00

Table 30: Distance matrix @ step 6

d_mat	1	2,9,3	5,15,12,4	6,8	7	10	11	13	14
1	0.00								
2,9,3	0.93	0.00							
5,15,12,4	5.69	5.47	0.00						
6,8	2.25	2.09	3.10	0.00					
7	3.56	3.50	2.44	1.20	0.00				
10	5.98	5.69	1.00	3.55	3.08	0.00			
11	4.98	4.84	0.88	2.44	1.83	1.56	0.00		
13	4.70	4.30	1.72	2.19	2.28	1.89	1.45	0.00	
14	3.61	3.34	2.31	1.47	1.57	2.45	1.84	1.68	0.00

Table 31: Distance matrix @ step 7

d_mat	1	2,9,3	5,15,12,4,11	6,8	7	10	13	14
1	0.00							
2,9,3	0.93	0.00						
5,15,12,4,11	4.98	4.84	0.00					
6,8	2.25	2.09	2.44	0.00				
7	3.56	3.50	1.83	1.20	0.00			
10	5.98	5.69	1.56	3.55	3.08	0.00		
13	4.70	4.30	1.45	2.19	2.28	1.89	0.00	
14	3.61	3.34	1.84	1.47	1.57	2.45	1.68	0.00

2,3,9 and 1 are closest to each other, so assign them to the same cluster. Now the partition is 1,2,3,9, 4,5,11,12,15, 6,8 and the rest are individual points. Update the distance matrix and it becomes Table 32.

Table 32: Distance matrix @ step 8

d_mat	2,9,3,1	5,15,12,4,11	6,8	7	10	13	14
2,9,3,1	0.00						
5,15,12,4,11	4.84	0.00					
6,8	2.09	2.44	0.00				
7	3.50	1.83	1.20	0.00			
10	5.69	1.56	3.55	3.08	0.00		
13	4.30	1.45	2.19	2.28	1.89	0.00	
14	3.34	1.84	1.47	1.57	2.45	1.68	0.00

6,8 and 7 are closest to each other, so assign them to the same cluster. Now the partition is 1,2,3,9, 4,5,11,12,15, 6,7,8 and the rest are individual points. Update the distance matrix and it becomes Table 33.

Table 33: Distance matrix @ step 9

d_mat	2,9,3,1	5,15,12,4,11	6,8,7	10	13	14
2,9,3,1	0.00					
5,15,12,4,11	4.84	0.00				
6,8,7	2.09	1.83	0.00			
10	5.69	1.56	3.08	0.00		
13	4.30	1.45	2.19	1.89	0.00	
14	3.34	1.84	1.47	2.45	1.68	0.00

4,5,11,12,15 and 13 are closest to each other, so assign them to the same cluster. Now the partition is 1,2,3,9, 4,5,11,12,13,15, 6,7,8 and the rest are individual points. Update the distance matrix and it becomes Table 34.
6,7,8 and 14 are closest to each other, so assign them to the same cluster.

Table 34: Distance matrix @ step 10

d_mat	2,9,3,1	5,15,12,4,11,13	6,8,7	10	14
2,9,3,1	0.00				
5,15,12,4,11,13	4.30	0.00			
6,8,7	2.09	1.83	0.00		
10	5.69	1.56	3.08	0.00	
14	3.34	1.68	1.47	2.45	0.00

Now the partition is 1,2,3,9, 4,5,11,12,13,15, 6,7,8,14 and the rest are individual points. Update the distance matrix and it becomes Table 35.

Table 35: Distance matrix @ step 11

d_mat	2,9,3,1	5,15,12,4,11,13	6,8,7,14	10
2,9,3,1	0.00			
5,15,12,4,11,13	4.30	0.00		
6,8,7,14	2.09	1.68	0.00	
10	5.69	1.56	2.45	0.00

4,5,11,12,13,15 and 10 are closest to each other, so assign them to the same cluster. Now the partition is 1,2,3,9, 4,5,10,11,12,13,15 and 6,7,8,14. Update the distance matrix and it becomes Table 36.

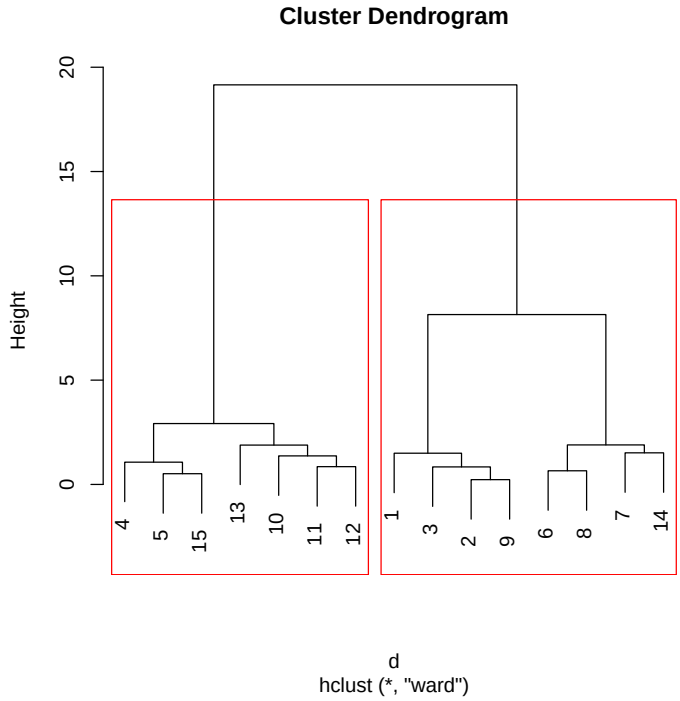
Table 36: Distance matrix @ step 12

d_mat	2,9,3,1	5,15,12,4,11,13,10	6,8,7,14
2,9,3,1	0.00		
5,15,12,4,11,13,10	4.30	0.00	
6,8,7,14	2.09	1.68	0.00

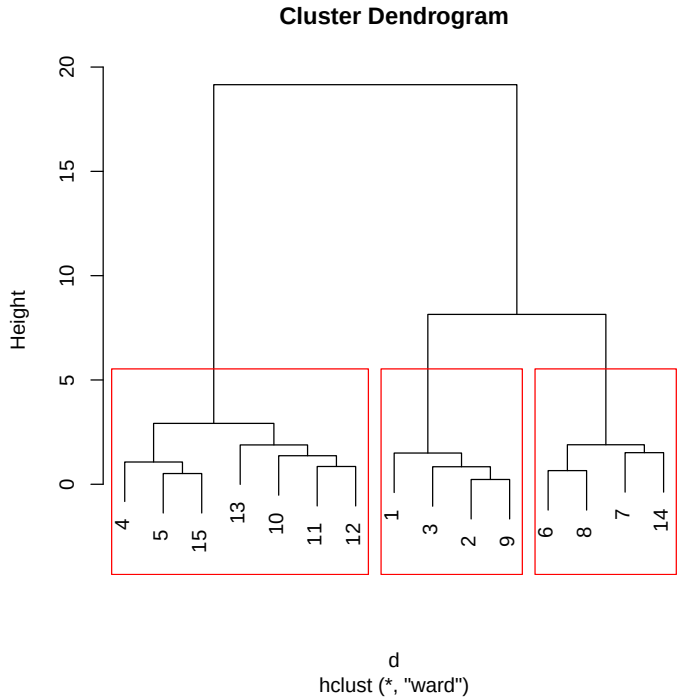
4,5,10,11,12,13,15 and 6,7,8,14 are closest to each other, so assign them to the same cluster. Now the partition is 1,2,3,9 and 4,5,6,7,8,10,11,12,13,14,15. Update the distance matrix and it becomes Table 37.
The final dendrogram with 2 clusters:

Table 37: Distance matrix @ step 13

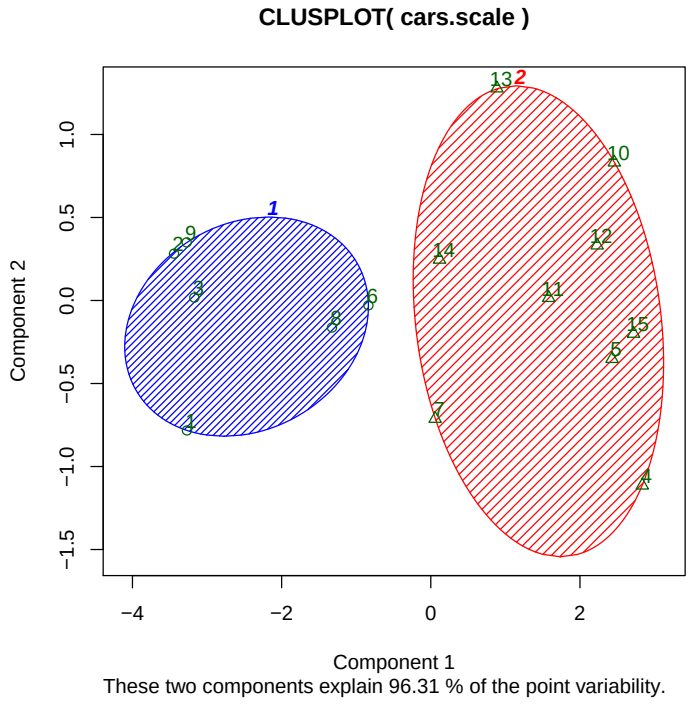
d_mat	2,9,3,1	5,15,12,4,11,13,10,6,8,7,14
2,9,3,1	0.00	
5,15,12,4,11,13,10,6,8,7,14	2.09	0.00

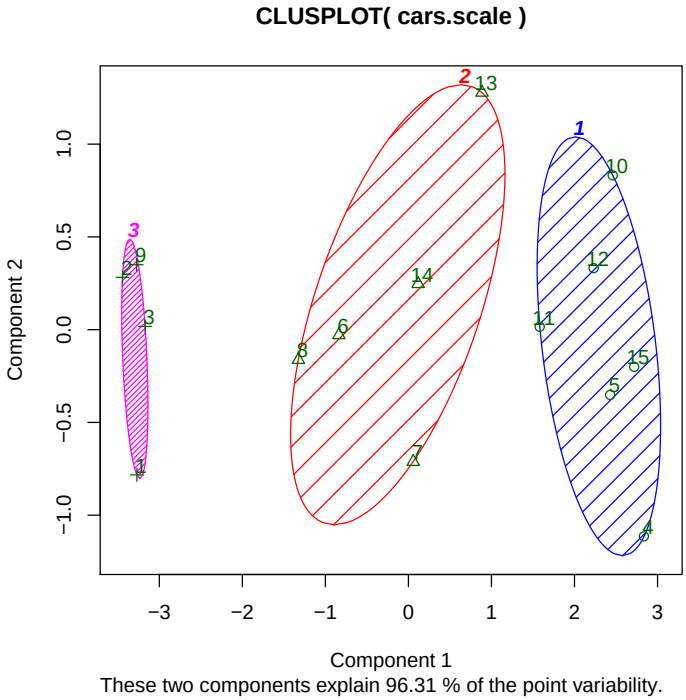


The final dendrogram with 3 clusters:



The difference from k-means partitioning is the belonging of 7, 14 for 2-cluster and 13 for 3-cluster.
Recap the k-means plots:





In the first plot, 7 and 14 are closer to 6, 8 rather than the 11, 12, etc. In the second plot, 13 is closer to 10 than 14. However, the goal of k-means is to minimize the within-cluster sum of squares (WCSS) instead of just to put closer points together. What's more is that k-means require an input of k. The parameter affect the clustering quality. That k can be obtained from the dendrogram. As shown in the two dendrograms, separating the points into only 2 clusters will make one of the cluster consisting of two groups relatively far from each other. From this point of view, hierarchical clustering algorithm is helpful to decide the optimal k for k-means.