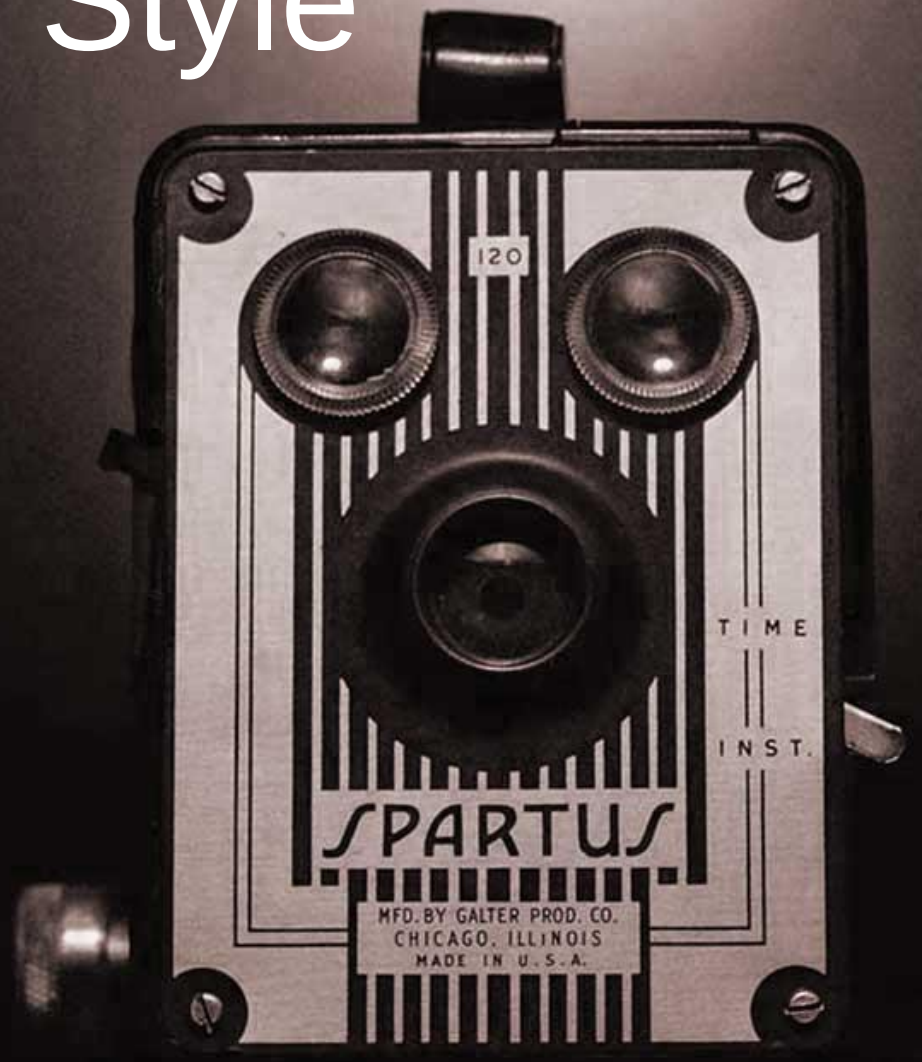
K-Means Clustering

Customer Movie Rating

Moh. Ridwan

Agenda

Style



01 Read The Data

Read customer movie rating data.

02 Data Preparation

Check all entries each column, handling outliers, and scale the data

03 Determine The Optimal K Value

Determine the optimal k value to determine the best model.

04 K-Mean Execution

Create a model with the optimal k value that has been found.

01. Read The Data

Read customer movie rating data and display top 6 data

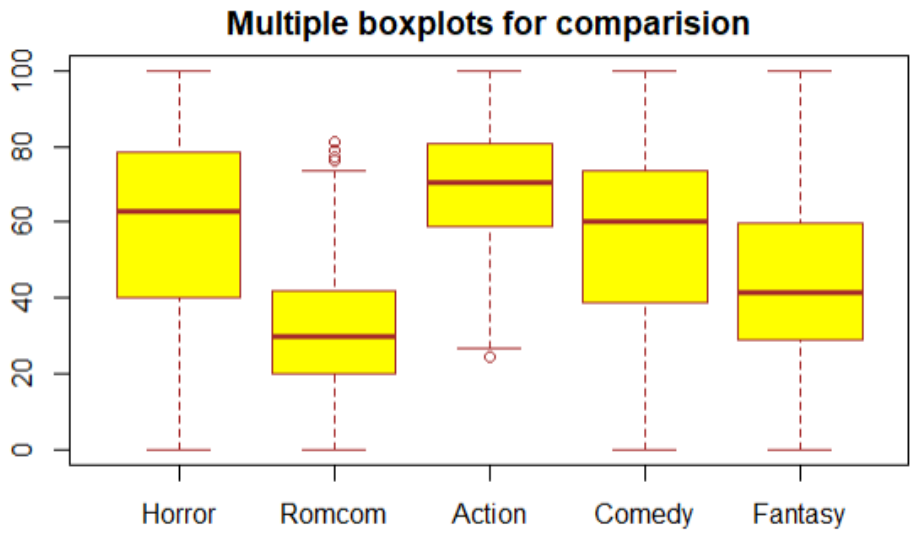
	Horror <dbl>	Romcom <dbl>	Action <dbl>	Comedy <dbl>	Fantasy <dbl>
1	72.5	29.9	68.6	40.7	57.9
2	82.2	45.3	76.5	17.4	67.7
3	70.0	44.0	65.1	53.7	37.8
4	99.1	21.0	77.9	25.4	40.3
5	84.0	0.0	68.1	49.8	40.0
6	70.2	55.0	97.2	48.1	40.5

02. Data Preparation

Summary data and check missing value

```
     Horror          Romcom          Action          Comedy     
 Min.   :  0.00   Min.   : 0.00   Min.   : 24.60   Min.   :  0.00  
 1st Qu.: 40.00   1st Qu.:19.90   1st Qu.: 58.75   1st Qu.: 38.50  
 Median : 62.80   Median :29.70   Median : 70.50   Median : 60.00  
 Mean   : 58.57   Mean   :31.25   Mean   : 68.84   Mean   : 56.52  
 3rd Qu.: 78.25   3rd Qu.:41.65   3rd Qu.: 80.55   3rd Qu.: 73.45  
 Max.   :100.00   Max.   :81.30   Max.   :100.00   Max.   :100.00  
    Fantasy      
 Min.   :  0.00  
 1st Qu.: 28.95  
 Median : 41.20  
 Mean   : 45.61  
 3rd Qu.: 59.85  
 Max.   :100.00  
```

Check outliers data with Multiple Boxplot



Handling outliers

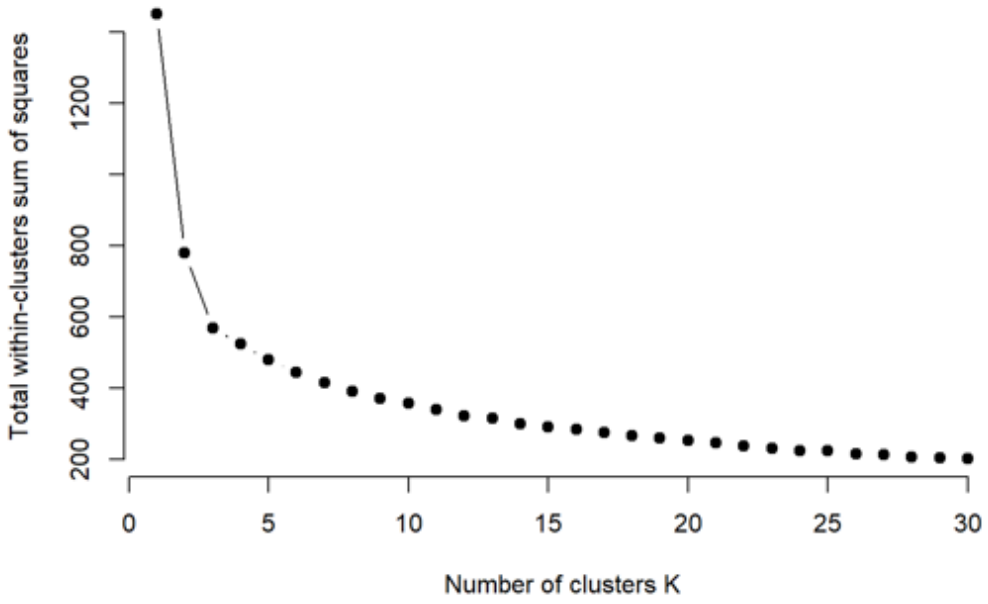
```
data$Romcom[is.na(data$Romcom)]<-mean(data$Romcom,na.rm=TRUE)
data$Action[is.na(data$Action)]<-mean(data$Action,na.rm=TRUE)

sum(is.na(data$Romcom))
sum(is.na(data$Action))
```

Scale the data

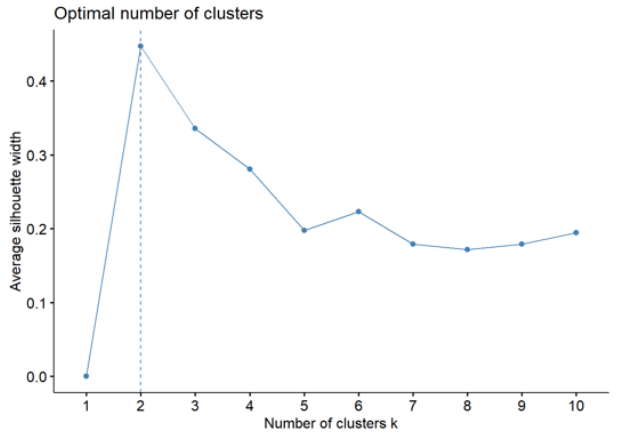
```
          Horror       Romcom       Action      Comedy     Fantasy
[1,] 0.5716500 -0.04299424 -0.02508091 -0.6986964  0.5587483
[2,] 0.9696365  0.91673283  0.47807457 -1.7277543  1.0042660
[3,] 0.4690762  0.83571691 -0.24799789 -0.1245439 -0.3550175
[4,] 1.6630357 -0.59764171  0.56724136 -1.3744297 -0.2413650
[5,] 1.0434897 -1.90636045 -0.05692619 -0.2967896 -0.2550033
[6,] 0.4772821  1.52123625  1.79646928 -0.3718711 -0.2322728
```

03. Determine The Optimum K Value

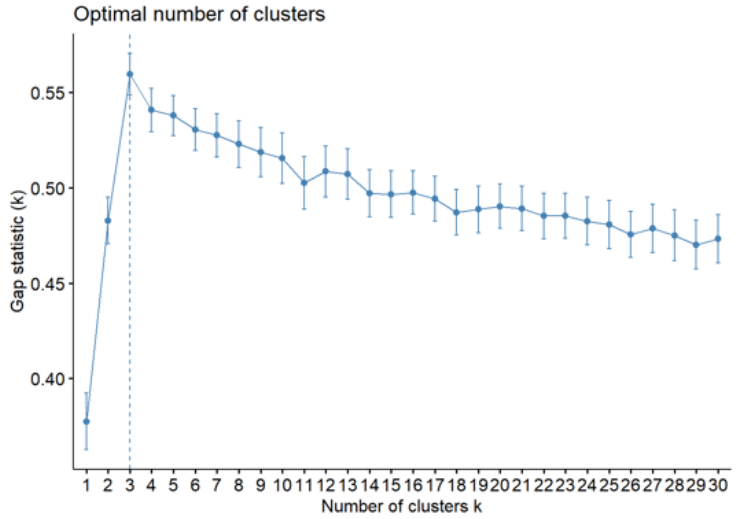
Total within-clusters sum of squares



Average silhouette method

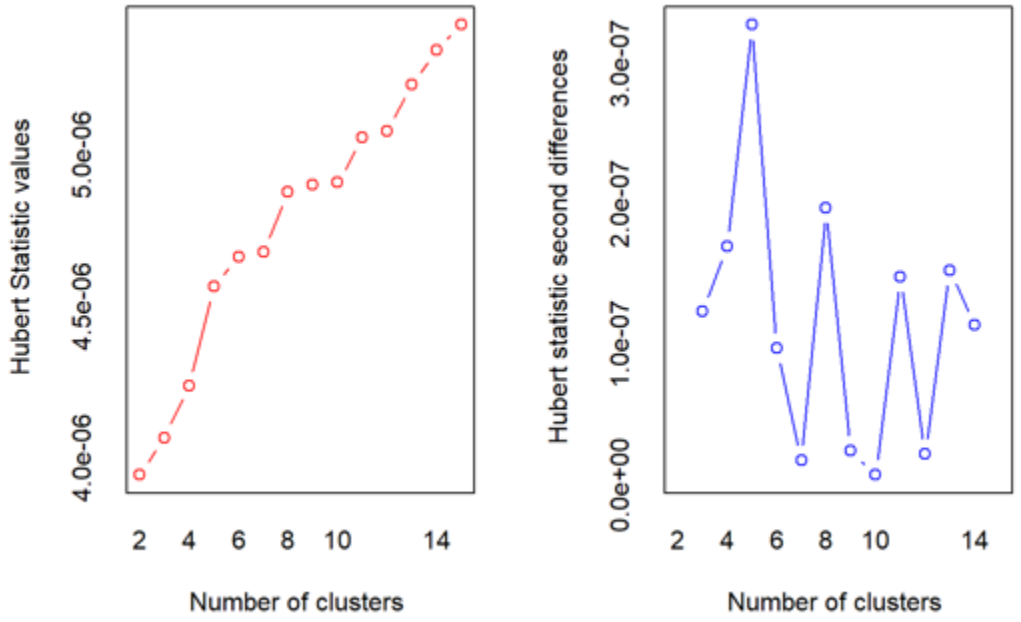


Gap statistic



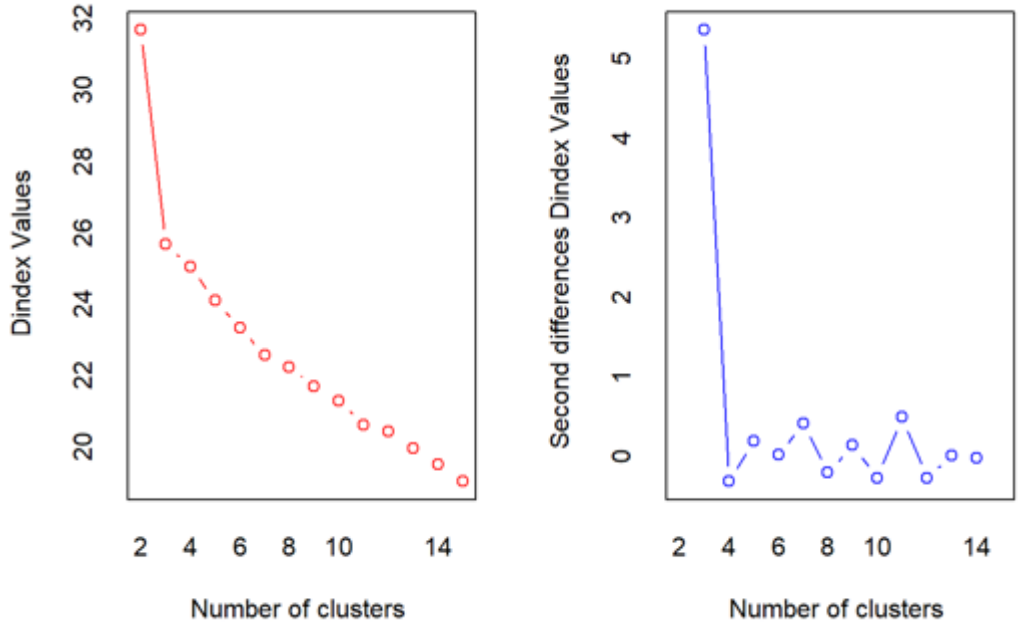
03. Determine The Optimum K Value

The optimal k value is based on several methods using NbClust



The Hubert index is a graphical method of determining the number of clusters. In the plot of Hubert index, we seek a significant knee that corresponds to a significant increase of the value of the measure i.e the significant peak in Hubert index second differences plot.

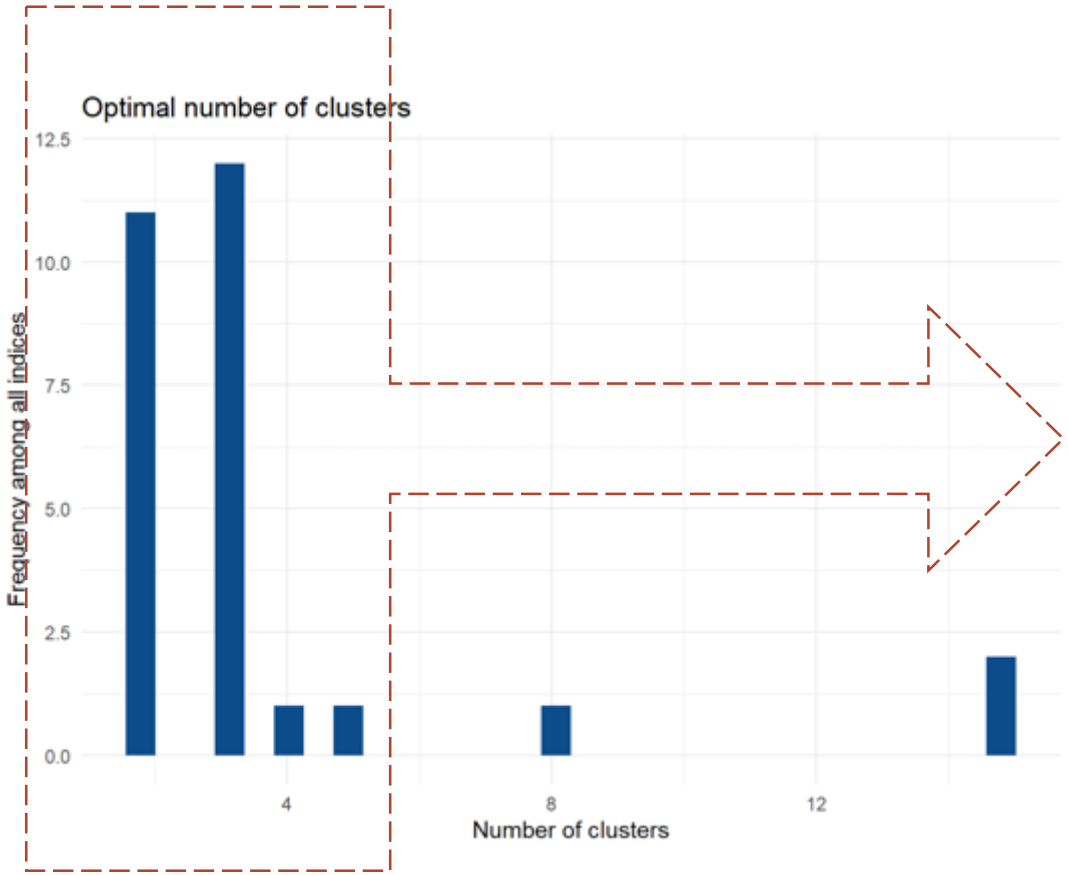
03. Determine The Optimum K Value

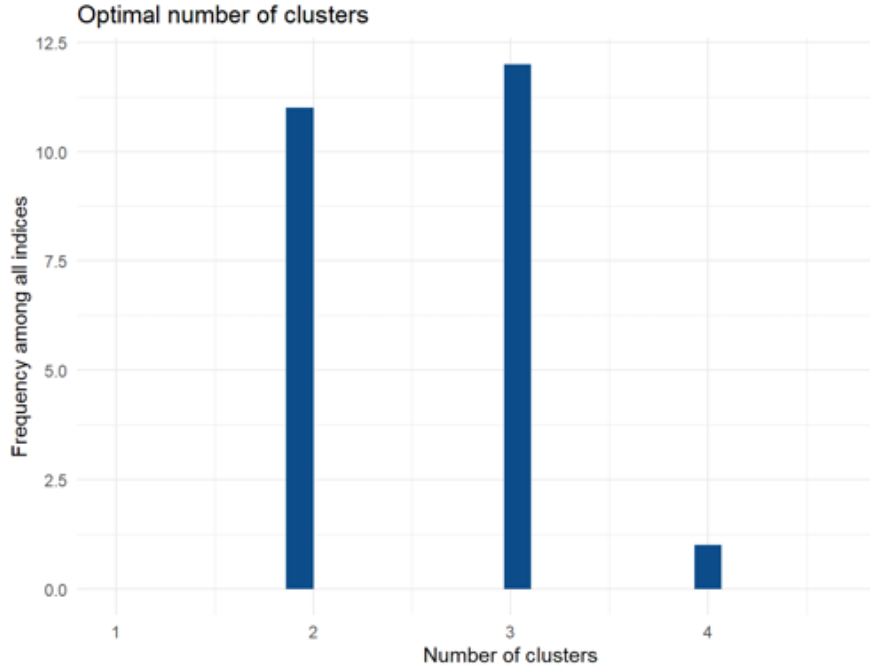


The D index is a graphical method of determining the number of clusters. In the plot of D index, we seek a significant knee (the significant peak in Dindex second differences plot) that corresponds to a significant increase of the value of the measure.

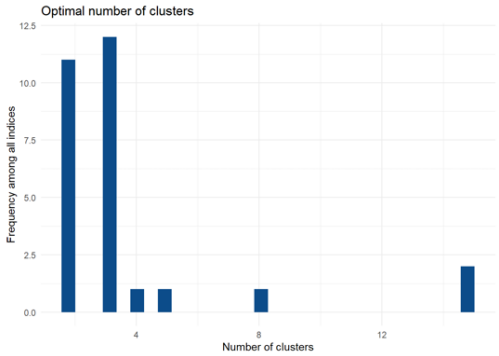
03. Determine The Optimum K Value

Make a plot of the optimal number of clusters from several test methods





03. Determine The Optimum K Value



Among all indices:

- 11 proposed 2 as the best number of clusters
- 12 proposed 3 as the best number of clusters
- 1 proposed 4 as the best number of clusters
- 1 proposed 5 as the best number of clusters
- 1 proposed 8 as the best number of clusters
- 2 proposed 15 as the best number of clusters

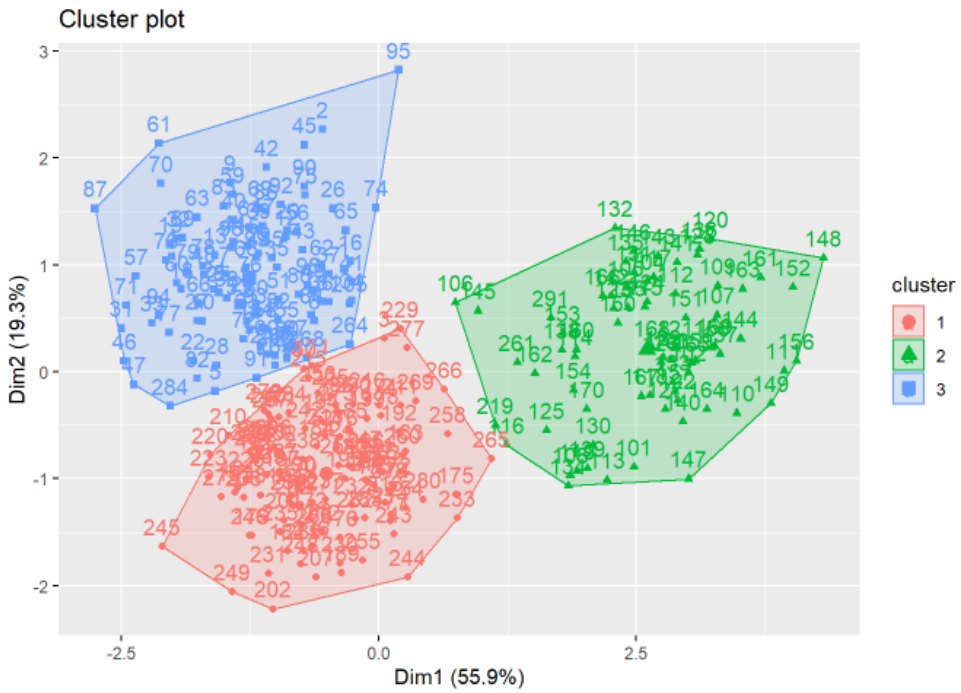
***** Conclusion *****

According to the majority rule, the best number of clusters is 3

04. K-Mean Execution

```
# Choose 123 as our random seed
set.seed(123)

# Cluster the players using kmeans with 3 clusters
cluster_solution <- kmeans(data_scale, centers = 3)
```

```
fviz_cluster(cluster_solution, data = data)
```



04. K-Mean Execution

Store the cluster assignments back into the clustering data frame object and display the top 6 data

Horror <dbl>	Romcom <dbl>	Action <dbl>	Comedy <dbl>	Fantasy <dbl>	cluster <fctr>
72.5	29.9	68.6	40.7	57.9	3
82.2	45.3	76.5	17.4	67.7	3
70.0	44.0	65.1	53.7	37.8	1
99.1	21.0	77.9	25.4	40.3	3
84.0	0.0	68.1	49.8	40.0	3
70.2	55.0	97.2	48.1	40.5	3

Look at the distribution of cluster assignments

```
  1   2   3
113  73 105
```


Thank You