LAU10-By-Shantal-Cruz.R

cs

2023-09-24

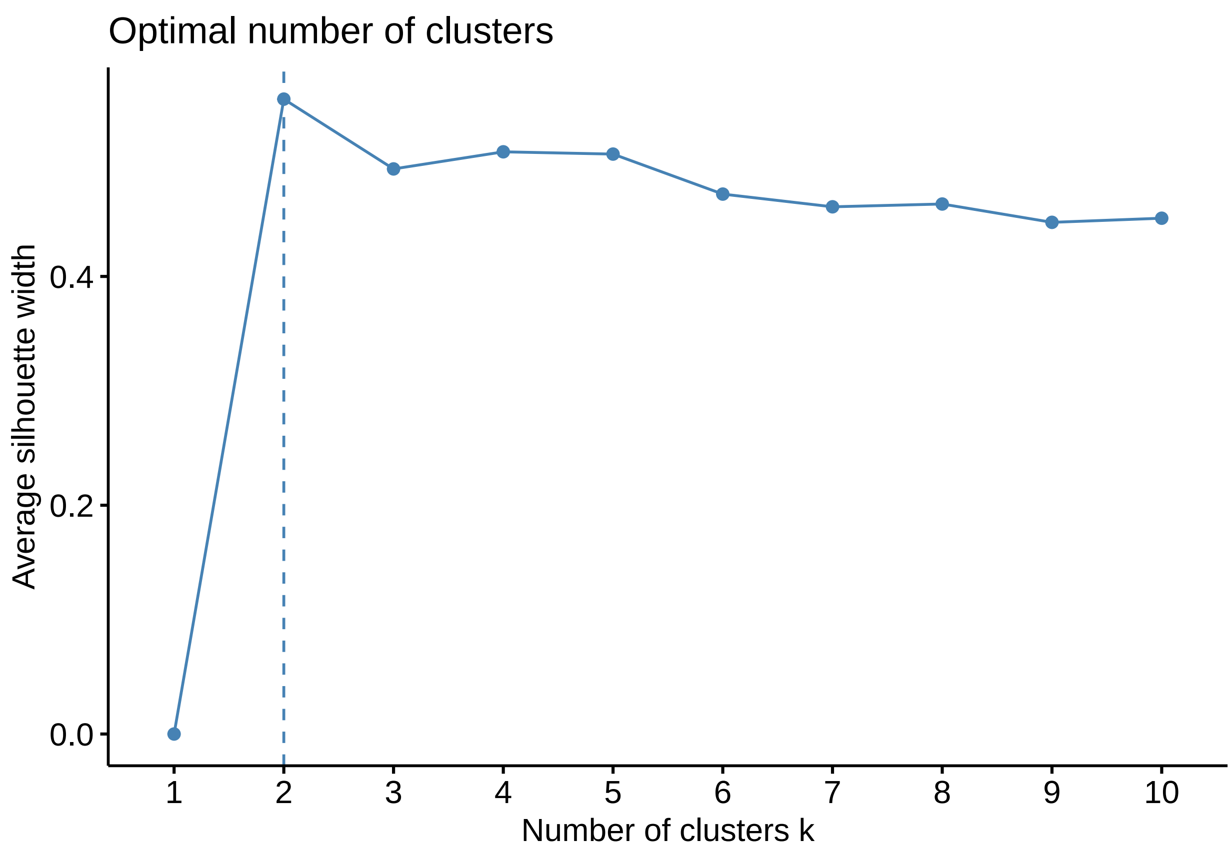
```r
library(readr)
data <- readr::read_csv("4063Midterm.csv")
```

```
## Rows: 1000 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Fname, Lname, gender, City
## dbl (9): ID, FamilyIncome, EdYears, FamilySize, Grocery, Cosmatics, MF, Boug...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# City Toronto only
myCity <- data[data$City == "Toronto", c("EdYears", "Cosmatics")]
# View(myCity)


# 1) Use a function such as factoextra::fviz_nbclust with silhouette method to identify the optimum num

# install.packages("factoextra")

library(factoextra)
```

```
## Loading required package: ggplot2
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(cluster)

# silhouette method
hc_s <- factoextra::fviz_nbclust(myCity, FUNcluster = hcut, method = "silhouette")
print(hc_s)
```




```r
hc_optimal_k <- 2
# 2) Use a function such as stats::hclust to build your hierarchical cluster model and use factoextra::f

# Hierarchical Clustering
hc <- stats::hclust(dist(myCity))
hc
```

```
##
## Call:
## stats::hclust(d = dist(myCity))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 255
```

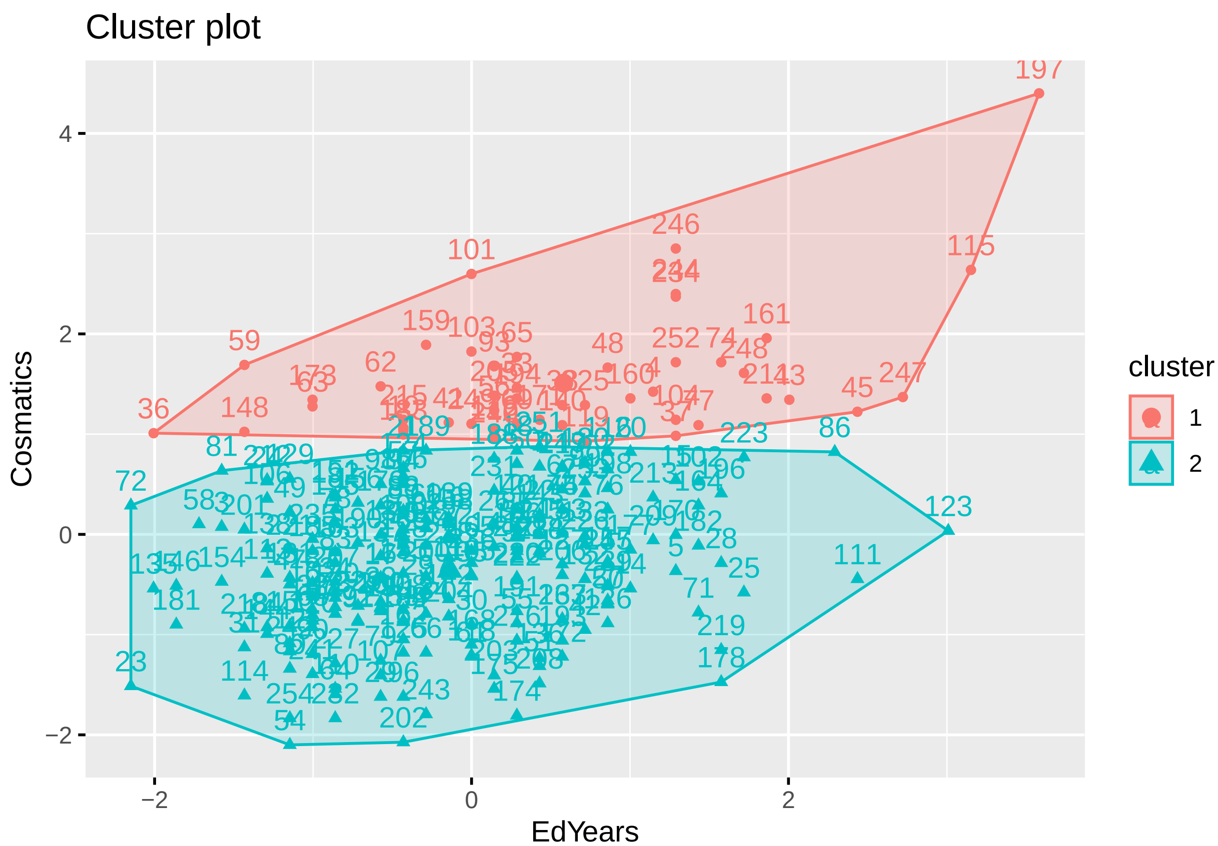
```r
cut_tree <- cutree(hc, k = hc_optimal_k)
cut_tree
```

```
##   [1] 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1
##  [38] 1 2 2 1 2 1 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 1 2 2 1 1 2 1 2 2 2 1 2 2 2 2 1
##  [75] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 2
## [112] 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1
## [149] 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 2 1 2 2
## [186] 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2
## [223] 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 1 1 1 1 2 2 1 2 2 2
```

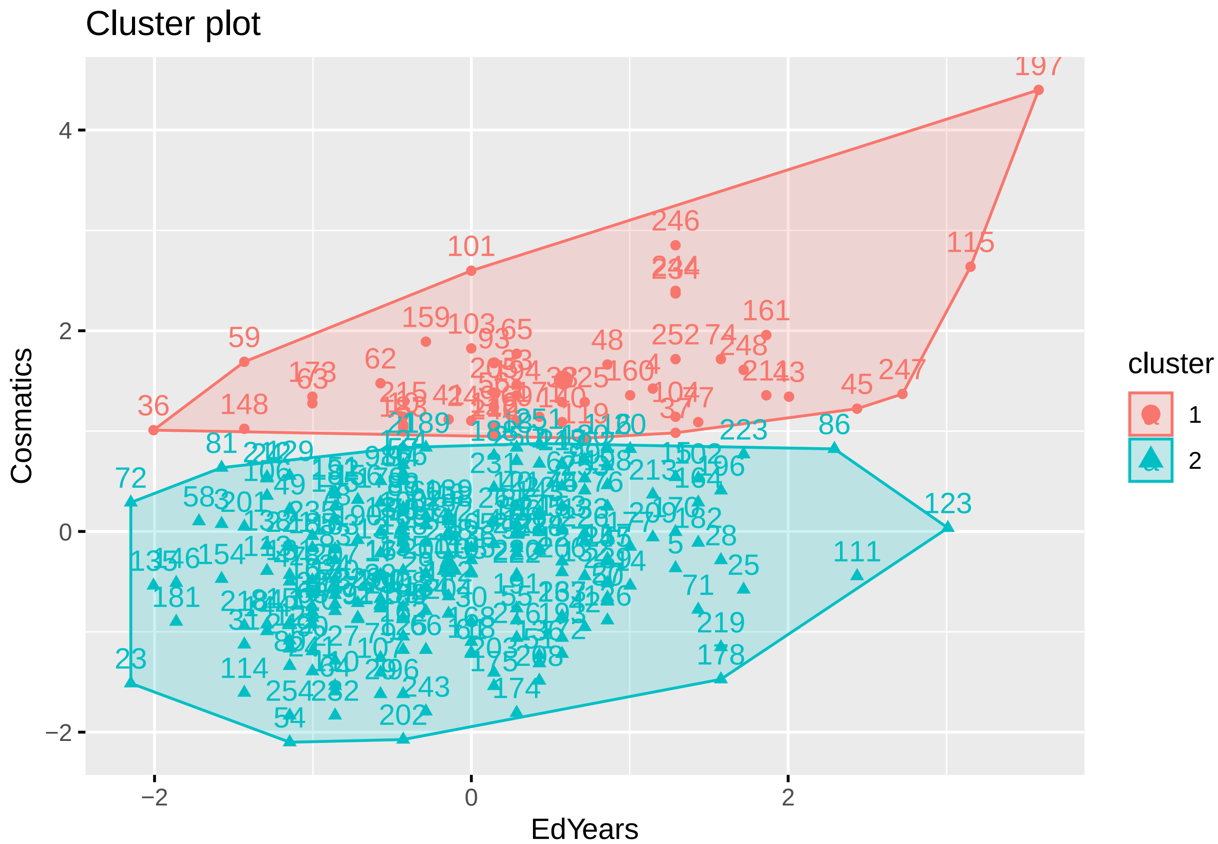
```r
# Visualize the dendrogram
dend <- as.dendrogram(hc)
dend
```

```
## 'dendrogram' with 2 branches and 255 members total, at height 488.1168
```

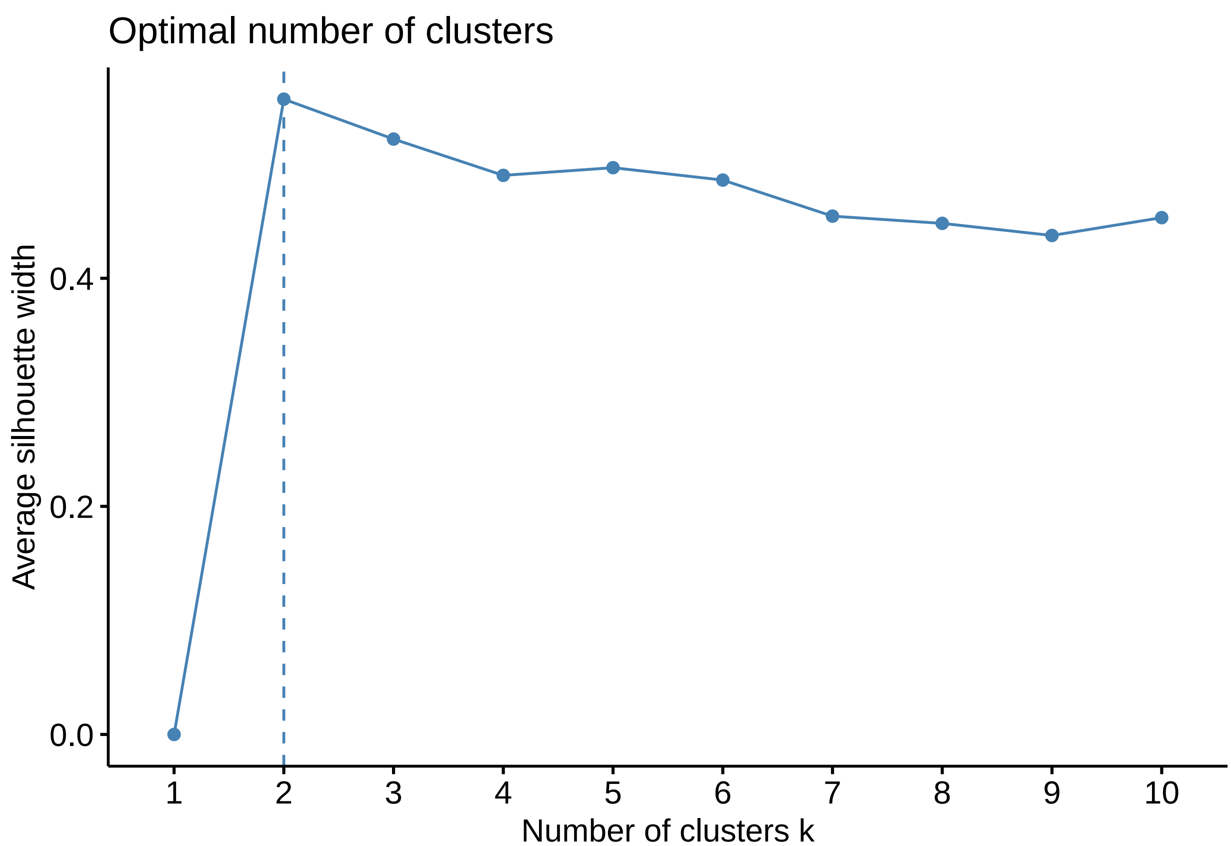
```r
# Create a clustering object using cut_tree
hc_cluster <- factoextra::fviz_cluster(list(data = myCity, cluster = cut_tree))
hc_cluster
```




```r
# Plot the clustering object
print(hc_cluster)
```




```r
# 3) Use  a function such as factoextra::fviz_nbclust with silhouette method to identify the optimum nu

# silhouette method
km_s <- factoextra::fviz_nbclust(myCity, FUNcluster = kmeans, method = "silhouette")
print(km_s)
```

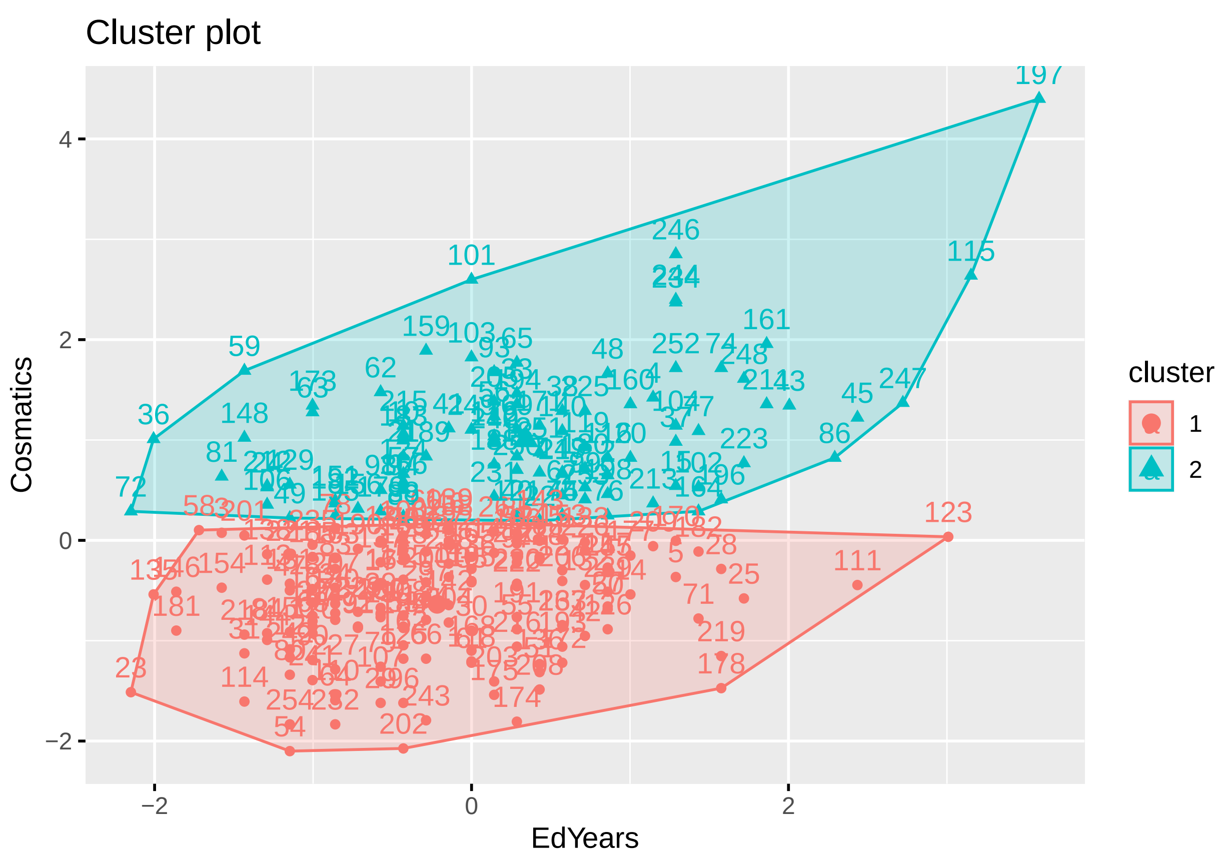



```
km_optimal_k <- 2

# 4) Use a function such as stats::Kmeans to build your Kmeans cluster model and use factoextra::fviz_c

# K-Means Clustering
km <- stats::kmeans(myCity, centers = km_optimal_k)
km
```

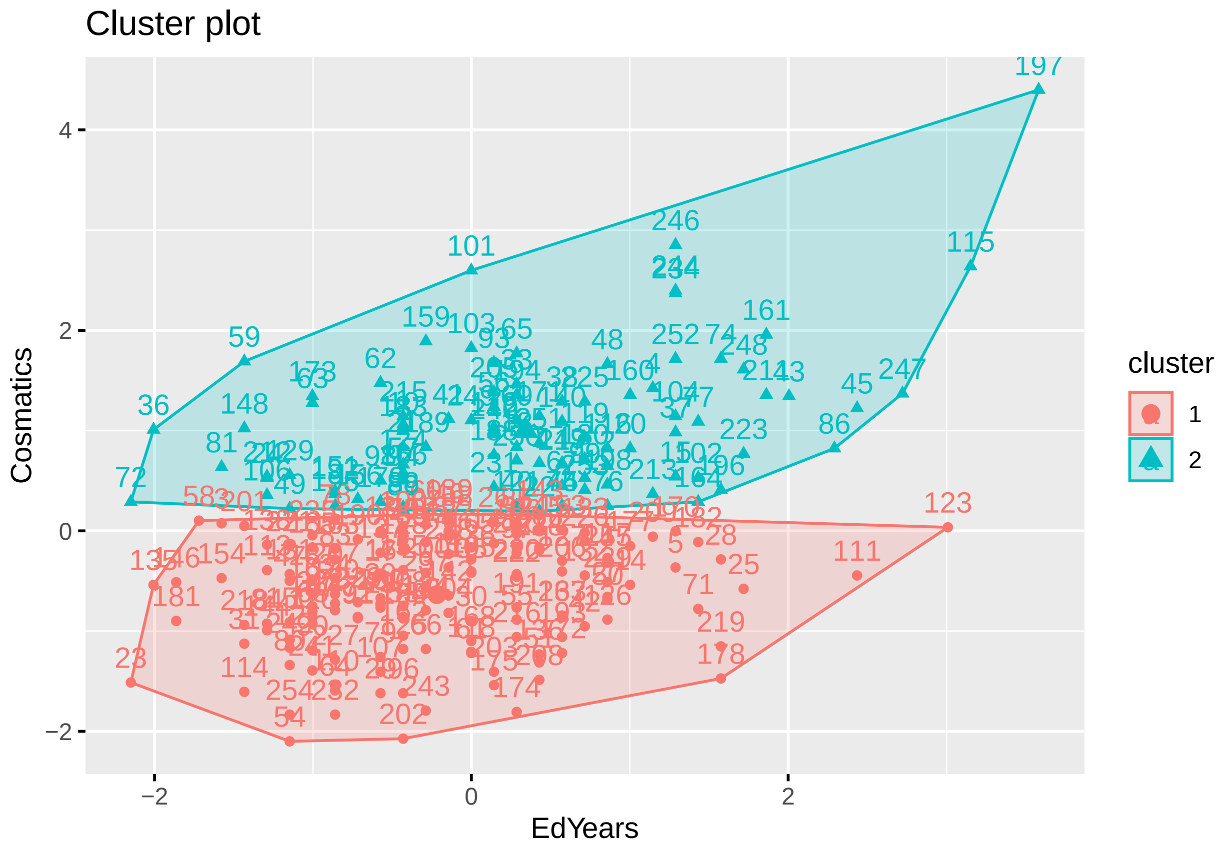
```
## K-means clustering with 2 clusters of sizes 156, 99
##
## Cluster means:
##    EdYears Cosmatics
## 1 13.48718  413.6987
## 2 17.39394  536.4444
##
## Clustering vector:
##   [1] 2 2 1 2 1 1 2 1 1 2 1 1 1 1 2 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 1 1 2 1 1 2 2
##  [38] 2 1 2 2 1 2 2 2 1 1 2 2 1 1 1 1 1 1 2 2 1 2 1 1 2 2 1 2 1 2 1 2 1 1 2 1 2
##  [75] 2 2 2 1 1 1 2 1 1 1 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 1 2 2 2 2 1 2 1 1 1 1 1
## [112] 2 1 1 2 2 1 1 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2
## [149] 2 1 2 1 1 1 1 2 1 1 2 2 2 1 1 2 1 2 1 1 1 1 2 1 2 1 1 2 1 1 2 2 1 1 2 1 1
## [186] 1 1 2 2 1 1 2 1 2 2 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 1 2 1 2 1 1 1 1 1
## [223] 2 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 1 1
##
## Within cluster sum of squares by cluster:
## [1] 259771.8 264838.1
```

```
##  (between_SS / total_SS =  63.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

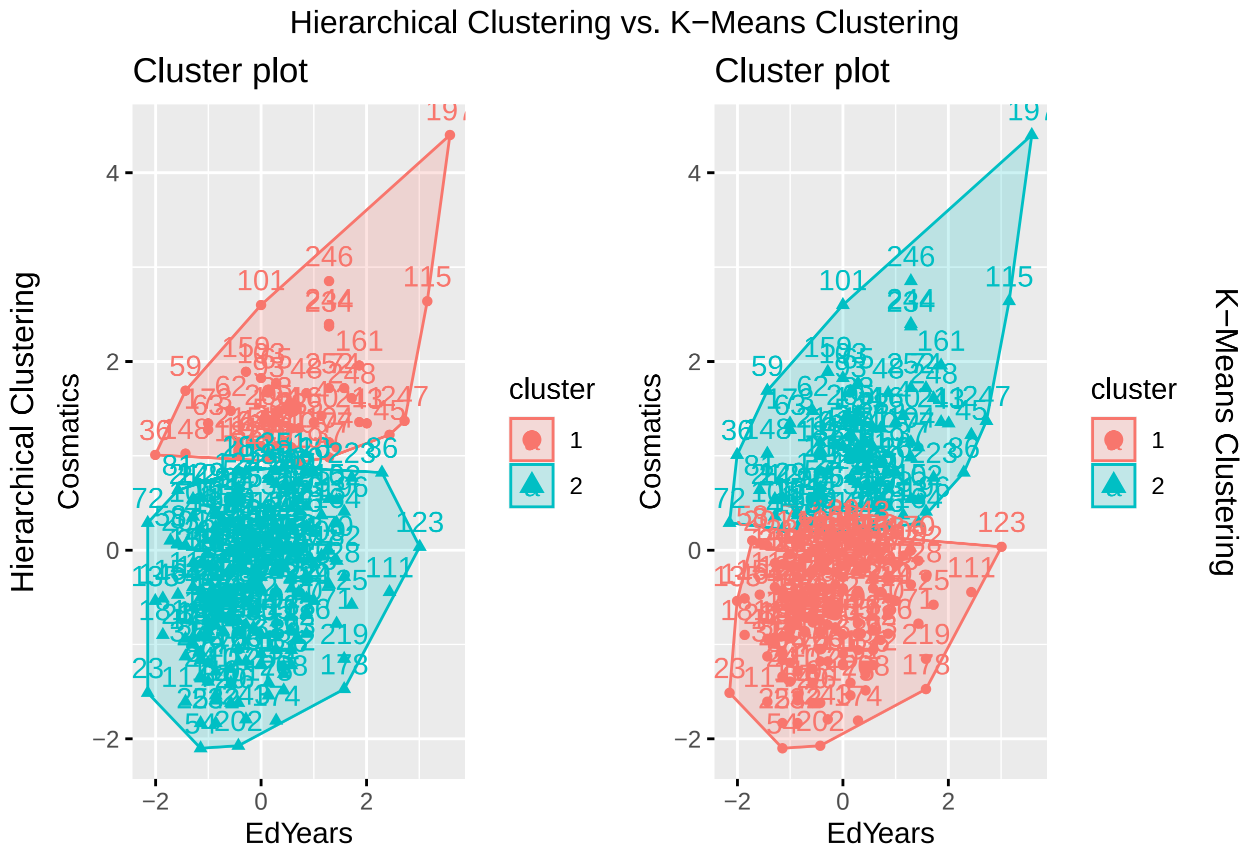
```
# Visualize the clustering object
km_cluster <- factoextra::fviz_cluster(list(data = myCity, cluster = km$cluster))
km_cluster
```




```
# Plot the clustering object
print(km_cluster)
```




```
# 5) Compare and contrast the visualizations of the clusters detected by two Machine Learning models. Wh
# put the two plots side by side
library(gridExtra)
grid.arrange(hc_cluster, km_cluster, ncol = 2, nrow = 1, top = "Hierarchical Clustering vs. K-Means Clus
```




```
print("To determine what clustering is better will depend on many factors. We can quantitatively use di
```

```
## [1] "To determine what clustering is better will depend on many factors. We can quantitatively use di
```