KMeans.R

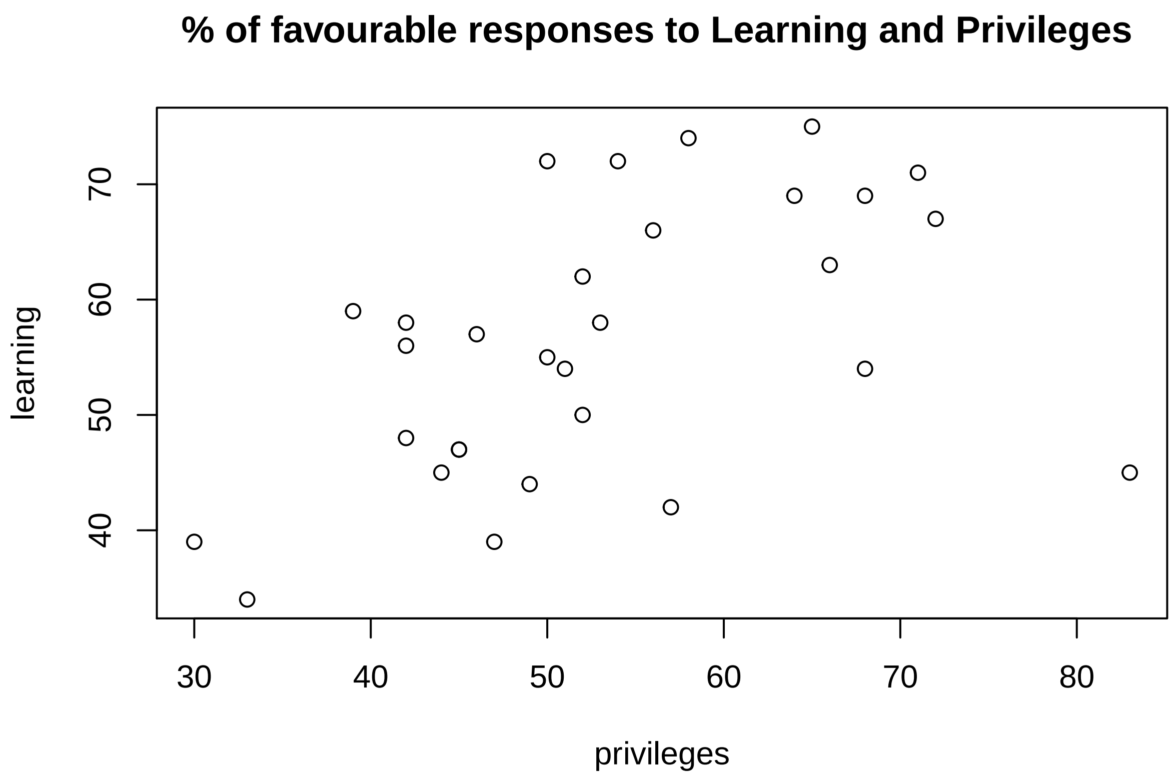
patriciamaya

2020-11-30

```r
#K-MEANS (unsupervised learning)
library(datasets)
?attitude

#all variables are numercial so we don't have to convert them to dummy variables
data <- attitude[, c(3,4)]
#we are using only 2 variables for learning purposes (see how clusters change)

plot(data, main="% of favourable responses to Learning and Privileges ")
```




```r
#no package needed for k-means
set.seed(7)
km1 <- kmeans(data, 2 , nstart= 100)
  #2 clusters
  #nstarts: num of times we want to run kmeans using different initial centrois
      #we repet kmeans 100 times and model return best one
km1
```

```
## K-means clustering with 2 clusters of sizes 17, 13
```

```
##
## Cluster means:
##   privileges learning
## 1   45.11765 48.94118
## 2   63.61538 66.07692
##
## Clustering vector:
##  [1] 1 1 2 1 2 1 1 1 2 1 1 1 1 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 1732.706 1920.000
##  (between_SS / total_SS =  56.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
#cluster means - centroids of each cluster
#WITHIN cluster sum of squares by cluster - variances within each cluster
    # -- want this to be 100%-- around (70-80% is good)
#SS-sum of squares
#total_SS = total variation of dataset
#BETWEEN_SS = variation between 2 clusters---- want this to be as large as possible

km1$cluster
```

```
##  [1] 1 1 2 1 2 1 1 1 2 1 1 1 1 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1
```

```r
km1$centers
```

```
##   privileges learning
## 1   45.11765 48.94118
## 2   63.61538 66.07692
```

```r
km1$withinss
```

```
## [1] 1732.706 1920.000
```

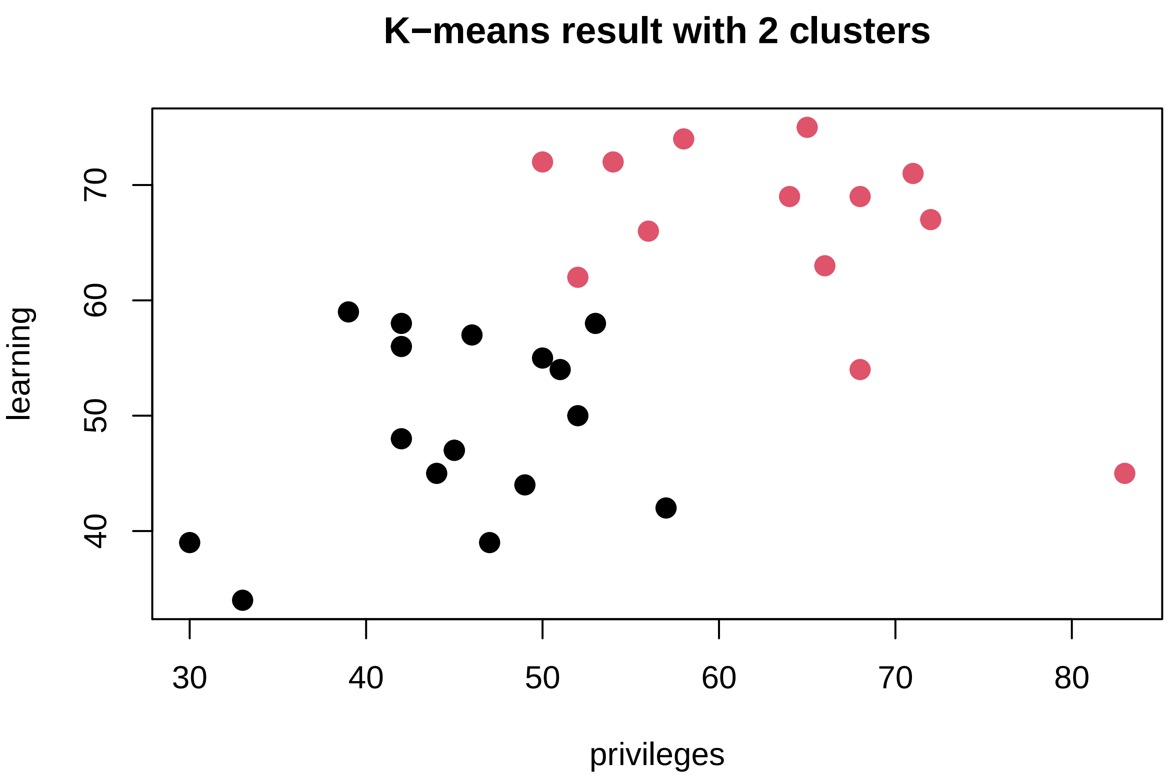
```r
km1$betweenss
```

```
## [1] 4683.727
```

```r
km1$size
```

```
## [1] 17 13
```

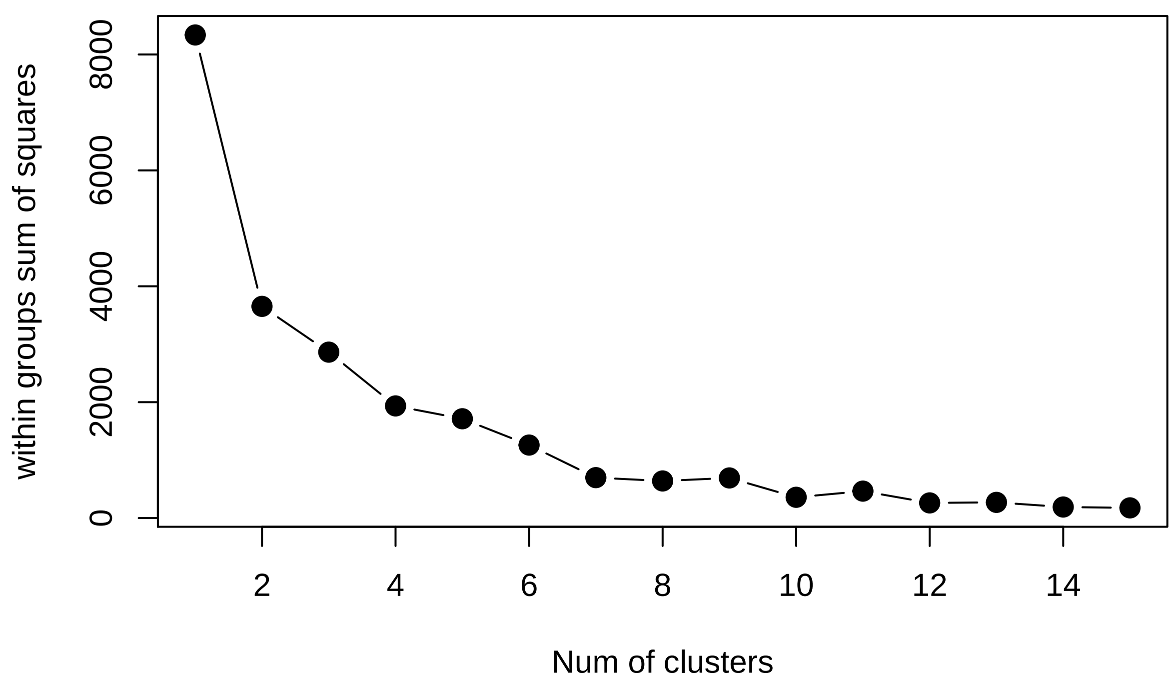
```r
plot(data, col=(km1$cluster), main="K-means result with 2 clusters", pch=20, cex=2)
```




```
#BEST VALUE OF K - scree plot

mydata <- data
wss <- (nrow(mydata) -1)*sum(apply(mydata, 2, var)) #total variance
for (i in 1:15)
  wss[i] <-  sum(kmeans(mydata, centers = i)$withinss)

plot(1:15, wss, type = "b", xlab= "Num of clusters", ylab="within groups sum of squares", pch=20, cex=2)
```

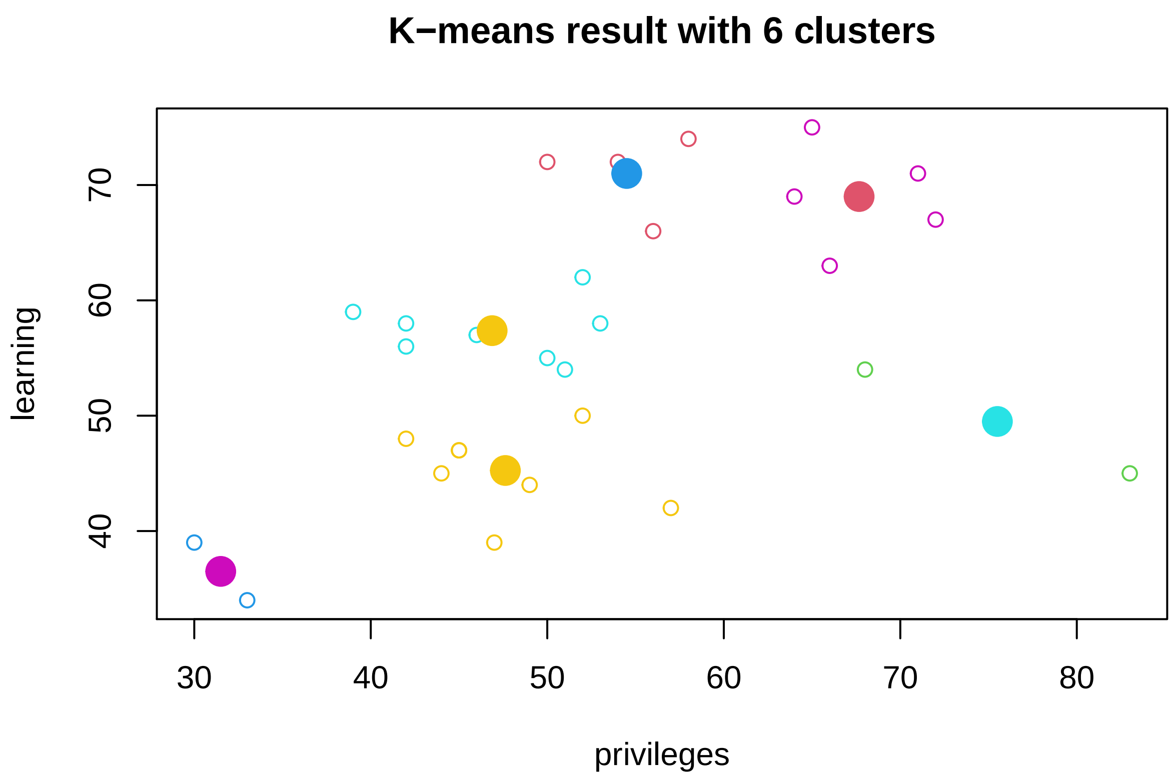


```r
#we can see after k=6 reduction of variation is not that significant
#thus, we can pick k=6 as the optimal num of k

set.seed(7)
km2 <- kmeans(data, 6, nstart=100)
km2
```

```
## K-means clustering with 6 clusters of sizes 4, 2, 2, 8, 6, 8
##
## Cluster means:
##   privileges learning
## 1   54.50000   71.000
## 2   75.50000   49.500
## 3   31.50000   36.500
## 4   46.87500   57.375
## 5   67.66667   69.000
## 6   47.62500   45.250
##
## Clustering vector:
##  [1] 3 4 5 6 1 6 4 4 5 6 4 6 6 2 1 1 5 5 4 2 3 4 6 4 6 5 1 6 5 4
##
## Within cluster sum of squares by cluster:
## [1]  71.0000 153.0000  17.0000 244.7500 133.3333 255.3750
##  (between_SS / total_SS =  89.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
#we can see "Within cluster sum of squares by cluster in" increased significantly
col<- (km2$cluster +1)
plot(data, col = col, main ="K-means result with 6 clusters")
points(km2$centers, col=col, pch=19, cex=2)
```




```
#bigger points are the centroids

#cluster 1 instances
cluster1 <- data[km2$cluster == 1 , ]
cluster1
```

```
##    privileges learning
## 5          56       66
## 15         54       72
## 16         50       72
## 27         58       74
```

```
#examine data for each of the clusters
#important variables-- variation is higher

#SILHOUETTE MEASURE -evaluate quality of clusters, works for any clustering method.
#a(i)= average distance of i from all the points in the same cluster
#b(i)= average distance of i from all the points in different clusters
#s(i) = b(i) - a(i) ) / max( a(i), b(i) )
  #[-1,1]. If close to 1, clustering is good -> clusters are NOT similar
  #if close  to 0, we can't tell if i is in the right cluster or not... points are too similar
  #if close to -1, a(i) is larger than b(i).. i is in the wrong cluster.
library(cluster)
# ?silhouette
avg_sil <- function(k)
{
  kmModel<- kmeans(data, centers=k, nstart=100)
  ss <- silhouette(kmModel$cluster, dist(data))
  mean(ss[,3])
}
```

```r
avg_sil(6) #avg silhouette using 6 clusters
```

```
## [1] 0.4368138
```