```
library("knitr")
library(ggplot2)
library(cluster)
library(factoextra)
```

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

```
library(gridExtra)
library(MASS)

data <- Boston
```

‘1. Select the numeric variables that you think are appropriate and useful. Use kmeans and Gaussian Mixture models.’

#Select the numeric variables

```
rm=data[,7]
dis=data[,9]
crim=data[,1]
medv=data[,14]
df <- data.frame(rm,dis,crim,medv)

X<- sample(c(rep(0, 0.7 * nrow(df)), rep(1, 0.3 * nrow(df))))
table(X)
```

```
## X
##   0   1
## 354 151
```

```
train <- df[X == 0, ]
test <- df[X== 1, ]
```

‘2. Split into train and test (70-30). Scale the data’

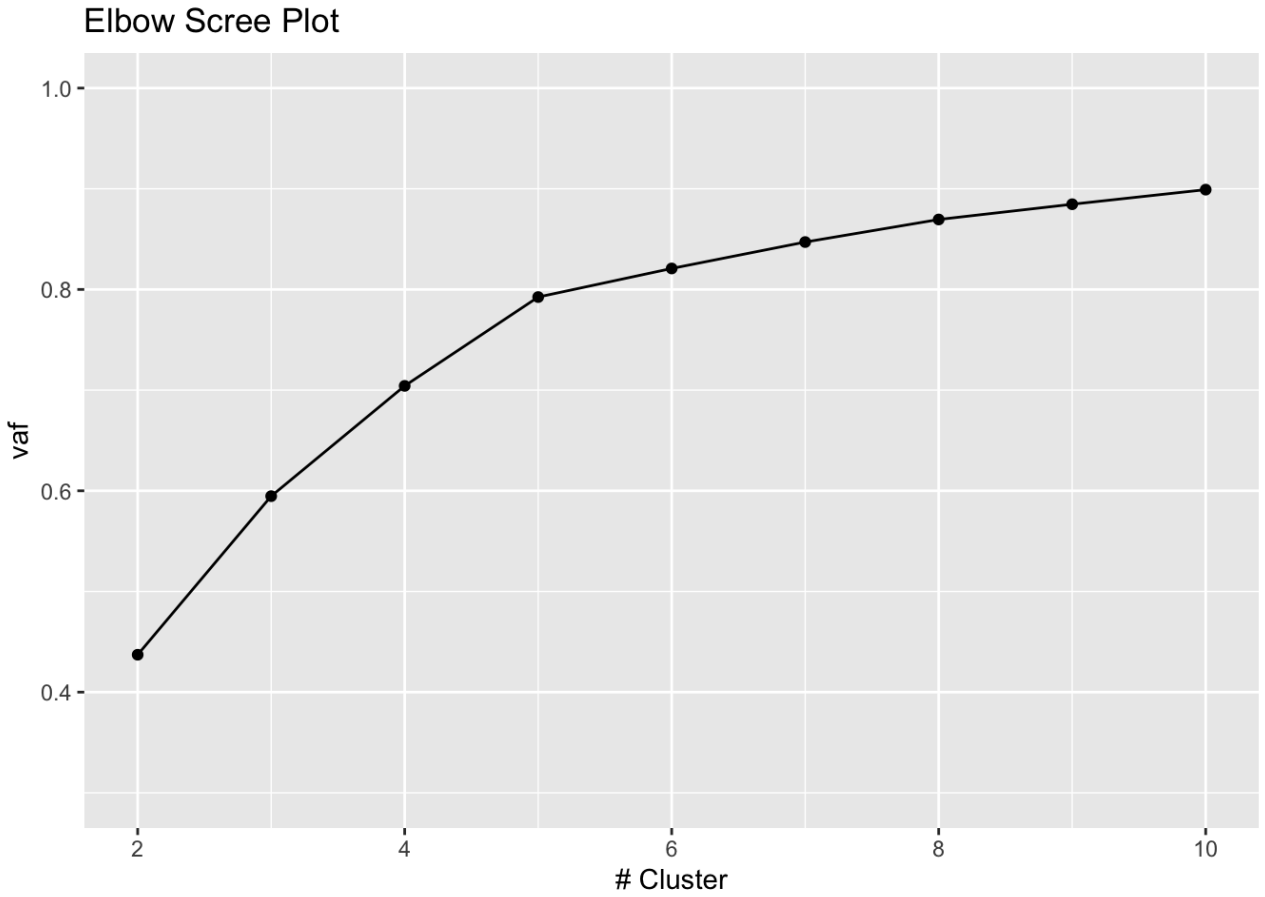
```
X.train.mean = colMeans(train)
X.train.sd   = sapply(train, sd)
X.train.scale  = scale(train, center=X.train.mean, scale=X.train.sd)
X.test.scale   = scale(test, center=X.train.mean, scale=X.train.sd) #scaling test by train parameters
```

‘3. Generate the K-means solution’

```
vaf = c()
centroid = c()
size = c()
num_clusters = seq(2, 10, by=1)
for (i in num_clusters){
  set.seed(123)
  kmeans_train = kmeans(X.train.scale,centers=i,nstart=70)
  vaf = append(vaf,1 - kmeans_train$tot.withinss / kmeans_train$totss)
  centroid = append(centroid, kmeans_train$centers)
  size = append(size,kmeans_train$size)
}
```

‘4. Perform a Scree test | 5. Show the scree plot.’

```
qplot(c(2:10), vaf) +
  geom_line() +
  xlab("# Cluster") +
  ylab("vaf") +
  ggtitle("Elbow Scree Plot") +
  ylim(0.3,1)
```



‘6.Choose 1 K-means solution (the best K from the last step) to retain from the many solutions that you have generated’

```
kmeans_train_single = kmeans(X.train.scale,centers=5,nstart=70)
kmeans_test_single = kmeans(X.train.scale,centers=kmeans_train_single$centers,nstart=70)

vaf_train = 1 - kmeans_train_single$tot.withinss / kmeans_train_single$totss
centroid_train = kmeans_train_single$centers
size_train = kmeans_train_single$size

vaf_test = 1 - kmeans_test_single$tot.withinss / kmeans_test_single$totss
centroid_test = kmeans_test_single$centers
size_test = kmeans_test_single$size


print(data.frame(vaf_train,vaf_test))
```

```
##   vaf_train  vaf_test
## 1 0.7924455 0.7924455
```

```
print(data.frame(centroid_train,centroid_test))
```

```
##            rm        dis       crim       medv        rm.1      dis.1
## 1  0.81676352  1.7144431  0.8290706 -0.8063701  0.81676352  1.7144431
## 2  0.09253265 -0.2625049 -0.2822728  2.0671479  0.09253265 -0.2625049
## 3  0.54523840 -0.5529085 -0.3322724 -0.3105061  0.54523840 -0.5529085
## 4  1.07489287  1.7144431  7.0926162 -1.6532732  1.07489287  1.7144431
## 5 -1.23443286 -0.5938779 -0.3893546  0.3060807 -1.23443286 -0.5938779
##       crim.1     medv.1
## 1  0.8290706 -0.8063701
## 2 -0.2822728  2.0671479
## 3 -0.3322724 -0.3105061
## 4  7.0926162 -1.6532732
## 5 -0.3893546  0.3060807
```

```
print(data.frame(size_train,size_test))
```

```
##   size_train size_test
## 1         80        80
## 2         36        36
## 3        122       122
## 4          4         4
## 5        113       113
```

‘7. Generate 3-5 Gaussian Mixtures (GM) | 8. Choose one solution & do interpretation’

```
require(mclust)
```

```
## Loading required package: mclust
```

```
## Package 'mclust' version 5.4.9
## Type 'citation("mclust")' for citing this R package in publications.
```

```
set.seed(123)
gm = Mclust(X.train.scale,G=3:5)
gm$bic #bic value of the selected model
```

```
## [1] -629.6974
```

```
gm$BIC #based on the table shown, the selected model is GMM with number of component=5 and
model=VEV
```

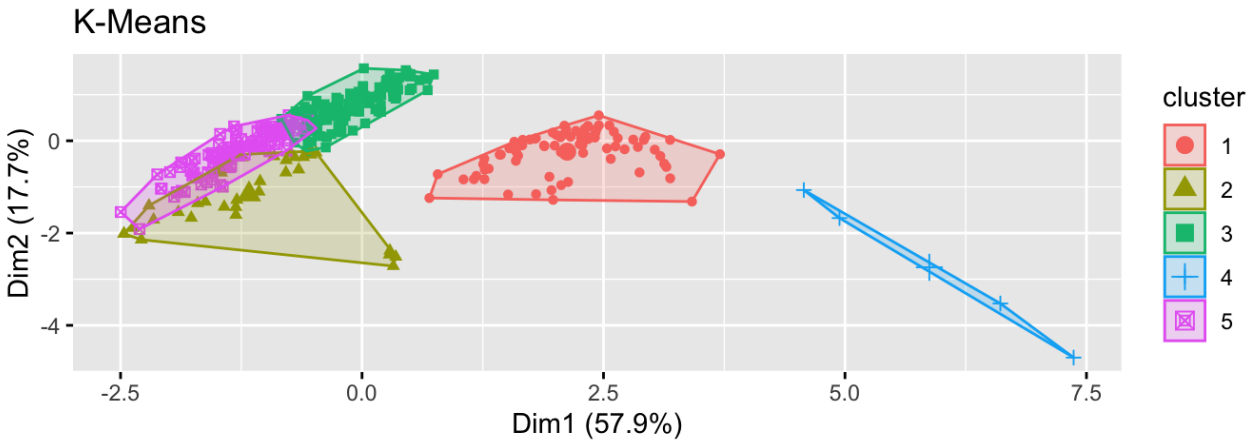
```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI EVI VVI       EEE       VEE EVE VVE
## 3 -3538.229 -3136.755 -2772.776 -2631.036  NA  NA -2778.710 -2641.596  NA  NA
## 4 -3362.093 -2797.828 -2764.171 -2130.191  NA  NA -2400.270 -2132.815  NA  NA
## 5 -3389.946 -2666.336 -2766.216 -2062.143  NA  NA -2398.431 -2069.427  NA  NA
##          EEV        VEV EVV VVV
## 3  -891.1271 -1483.5843  NA  NA
## 4 -1463.6682  -926.9720  NA  NA
## 5  -879.1715  -629.6974  NA  NA
##
## Top 3 models based on the BIC criterion:
##     VEV,5     EEV,5     EEV,3
## -629.6974 -879.1715 -891.1271
```

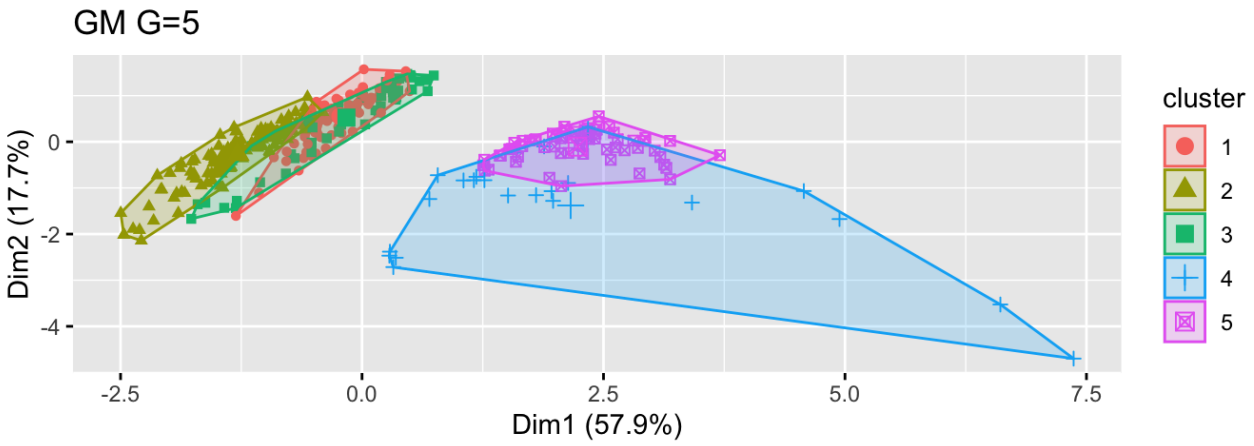
```
summary(gm) #most of the data are clustered in cluster 3 (51.23%), and the least is in clu
ster 3 (2.25%)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 5 components:
##
##  log-likelihood   n df       BIC       ICL
##       -132.8131 355 62 -629.6974 -656.9259
##
## Clustering table:
##   1   2   3   4   5
##  79 130  58  24  64
```

‘9. Build a GM model with the best components on train data and compare it with the train KMeans solution from an interpretability perspective’

```
p1 = fviz_cluster(kmeans_train_single,geom = "point", data=X.train.scale) + ggtitle("K-Mea
ns")
p2 = fviz_cluster(gm,geom = "point", data=X.train.scale) + ggtitle("GM G=5")
grid.arrange(p1, p2, nrow = 2)
```






```
kmeans_train_single$centers
```

```
##            rm        dis       crim       medv
## 1  0.81676352  1.7144431  0.8290706 -0.8063701
## 2  0.09253265 -0.2625049 -0.2822728  2.0671479
## 3  0.54523840 -0.5529085 -0.3322724 -0.3105061
## 4  1.07489287  1.7144431  7.0926162 -1.6532732
## 5 -1.23443286 -0.5938779 -0.3893546  0.3060807
```

```
gm$parameters$mean
```

```
##              [,1]       [,2]         [,3]        [,4]       [,5]
## rm    0.472893158 -1.1098339  0.614806667  0.57892501  0.9418273
## dis  -0.476966647 -0.6177760 -0.561053700  1.71444309  1.7144431
## crim -0.380846804 -0.3943492 -0.257639494  1.81207098  0.8004588
## medv -0.006412534  0.4532668  0.004910459 -0.07263518 -0.9207668
```

‘10. Summarize results and interpret the clusters/segments you choose as your final solution.’

```
kmeans_train_single$centers
```

```
##            rm        dis       crim       medv
## 1  0.81676352  1.7144431  0.8290706 -0.8063701
## 2  0.09253265 -0.2625049 -0.2822728  2.0671479
## 3  0.54523840 -0.5529085 -0.3322724 -0.3105061
## 4  1.07489287  1.7144431  7.0926162 -1.6532732
## 5 -1.23443286 -0.5938779 -0.3893546  0.3060807
```

```
kmeans_train_single$size
```

```
## [1]  80  36 122   4 113
```

```
gm$parameters$mean
```

```
##              [,1]       [,2]         [,3]        [,4]       [,5]
## rm    0.472893158 -1.1098339  0.614806667  0.57892501  0.9418273
## dis  -0.476966647 -0.6177760 -0.561053700  1.71444309  1.7144431
## crim -0.380846804 -0.3943492 -0.257639494  1.81207098  0.8004588
## medv -0.006412534  0.4532668  0.004910459 -0.07263518 -0.9207668
```

```
summary(gm)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 5 components:
##
##  log-likelihood   n df       BIC       ICL
##       -132.8131 355 62 -629.6974 -656.9259
##
## Clustering table:
##   1   2   3   4   5
##  79 130  58  24  64
```

'All interpretations are available in pdf document'