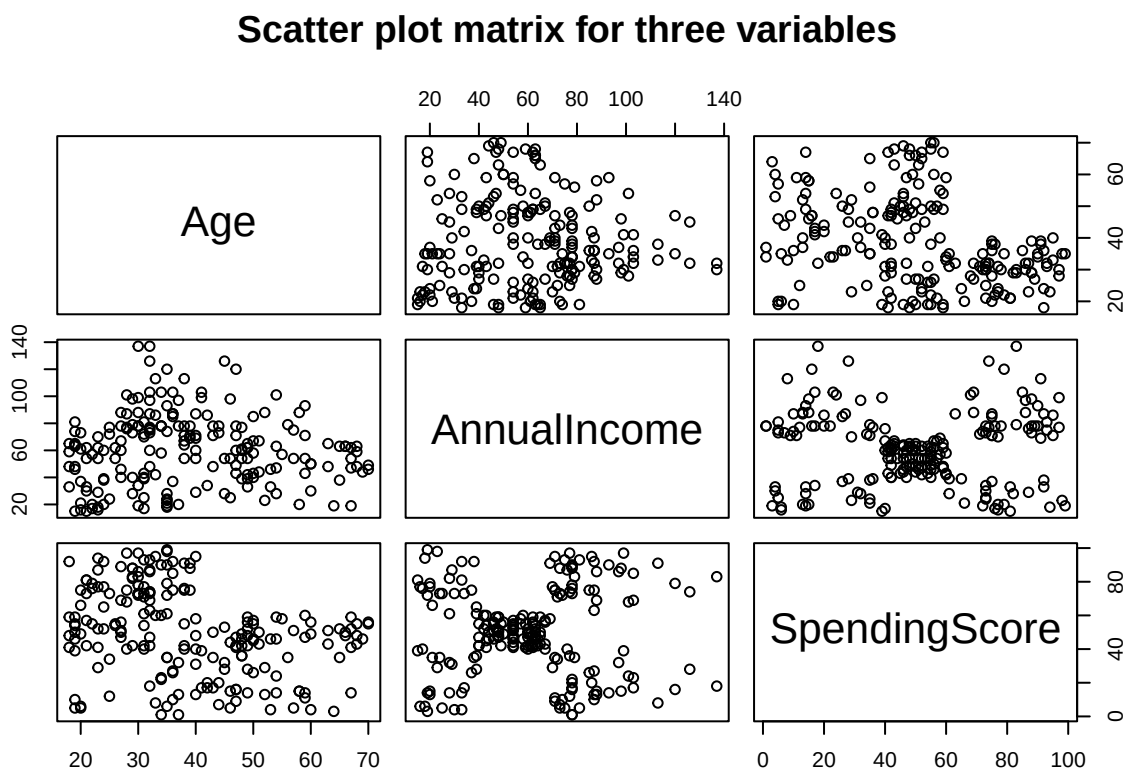
Cluster Analysis - Mall Customers data

Sanjaya Mananage

Scatterplot matrix of variables

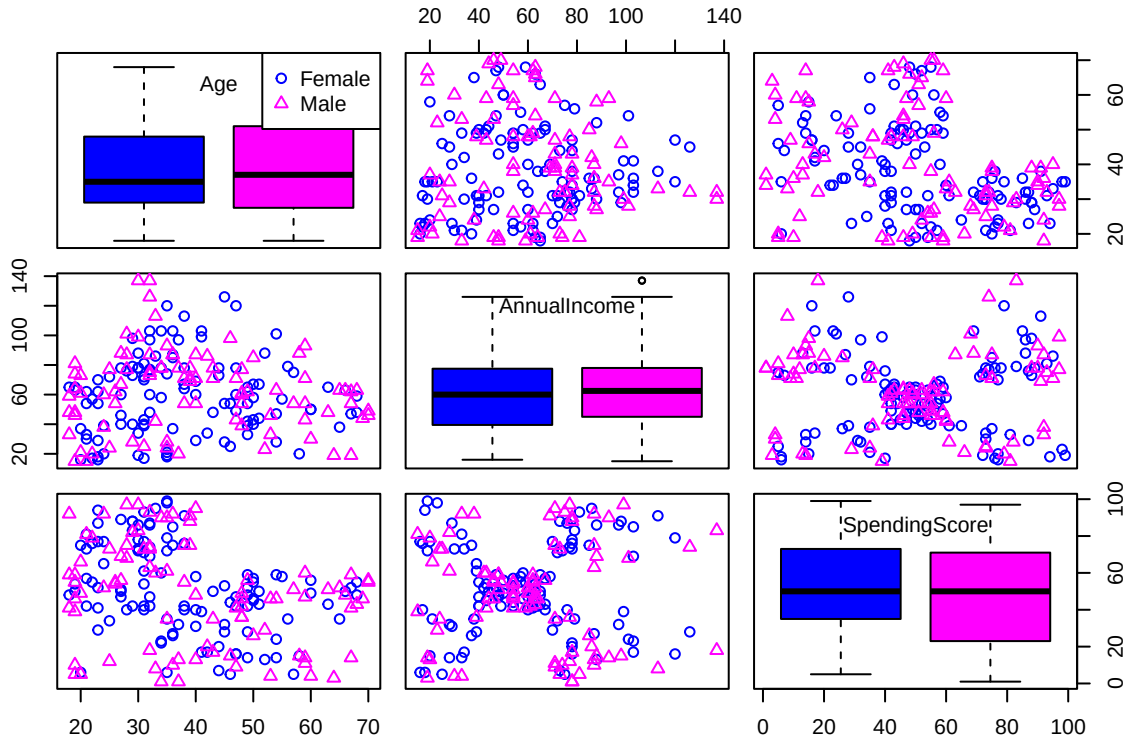
```
mall_customers<-read.csv("Mall_Customers.csv")
pairs(mall_customers[,3:5],main="Scatter plot matrix for three variables")
```




```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
scatterplotMatrix(mall_customers[,3:5], smooth= FALSE, regLine = F,
                  diagonal = list(method= "boxplot"),
                  groups =mall_customers$Gender ,cex.labels = 1)
```



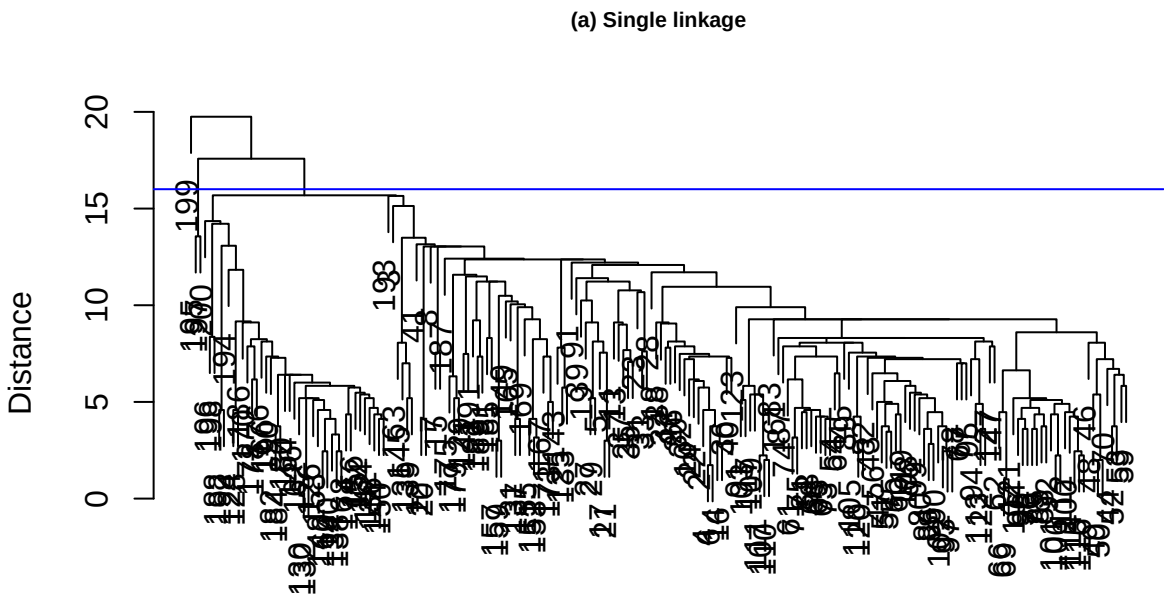
We cannot find a specific pattern on scatterplot matrix but some scatters are gathered around a point and others were spread for each pair of variables.

1. Hierarchical clustering

Perform hierarchical clustering with single, complete, and average linkage. Decide on the number of clusters using a cutpoint.

Hierarchical clustering using single linkage method

```
#dendogram from hierarchical clustering
plot(hclust(dist(mall_customers[,3:5]),method="single"),
     labels=row.names(mall_customers[,3:5]),ylab="Distance", main="(a) Single linkage",cex.main=0.7)
#cutting the dendogram at height 16 to get various clusters formed at that point.
abline(h=16,col="blue")
```




```
cluster1<-cutree(hclust(dist(mall_customers[,3:5]),method="single"),h=16)
max(cluster1)
```

```
## [1] 3
```

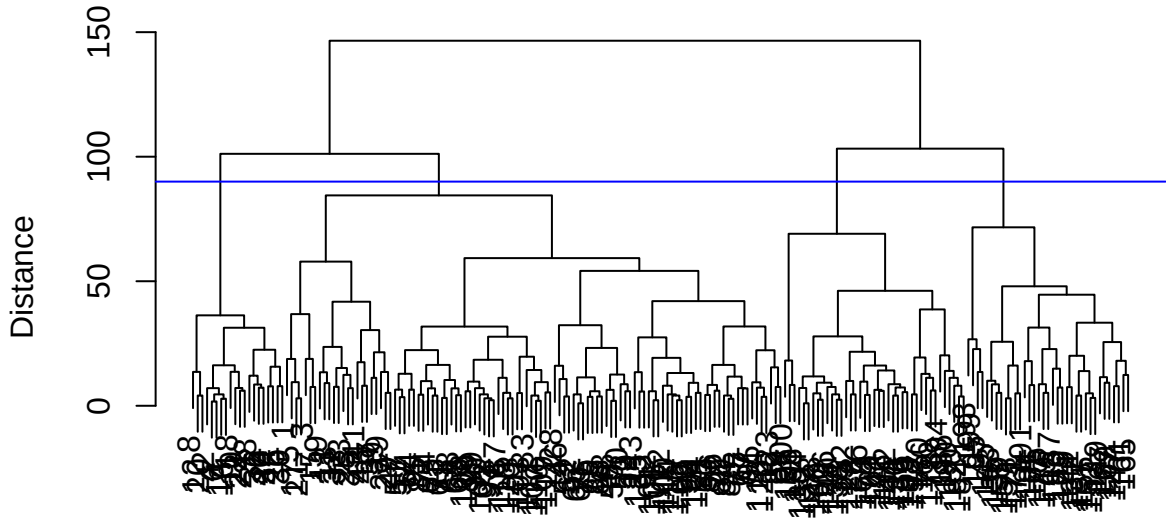
```
# Mean vectors for each cluster
cluster.mean1<-lapply(1:3,function(nc) {colMeans(mall_customers[cluster1==nc,3:5])})
cluster.mean1
```

```
## [[1]]
##           Age  AnnualIncome SpendingScore
##      38.81218      59.53807      50.64975
##
## [[2]]
##           Age  AnnualIncome SpendingScore
##            46           123            22
##
## [[3]]
##           Age  AnnualIncome SpendingScore
##            32           137            18
```

Hierarchical clustering using complete linkage method

```
plot(hclust(dist(mall_customers[,3:5]),method="complete"),
     labels=row.names(mall_customers[,3:5]),ylab="Distance", main="(b) Complete linkage",cex.main=0.7)

#cutting the dendogram at height 90 to get various clusters formed at that point.
abline(h=90,col="blue")
```

(b) Complete linkage



dist(mall_customers[, 3:5])
hclust (*, "complete")

```
cluster2<-cutree(hclust(dist(mall_customers[,3:5]),method="complete"),h=90)
max(cluster2)
```

```
## [1] 4
```

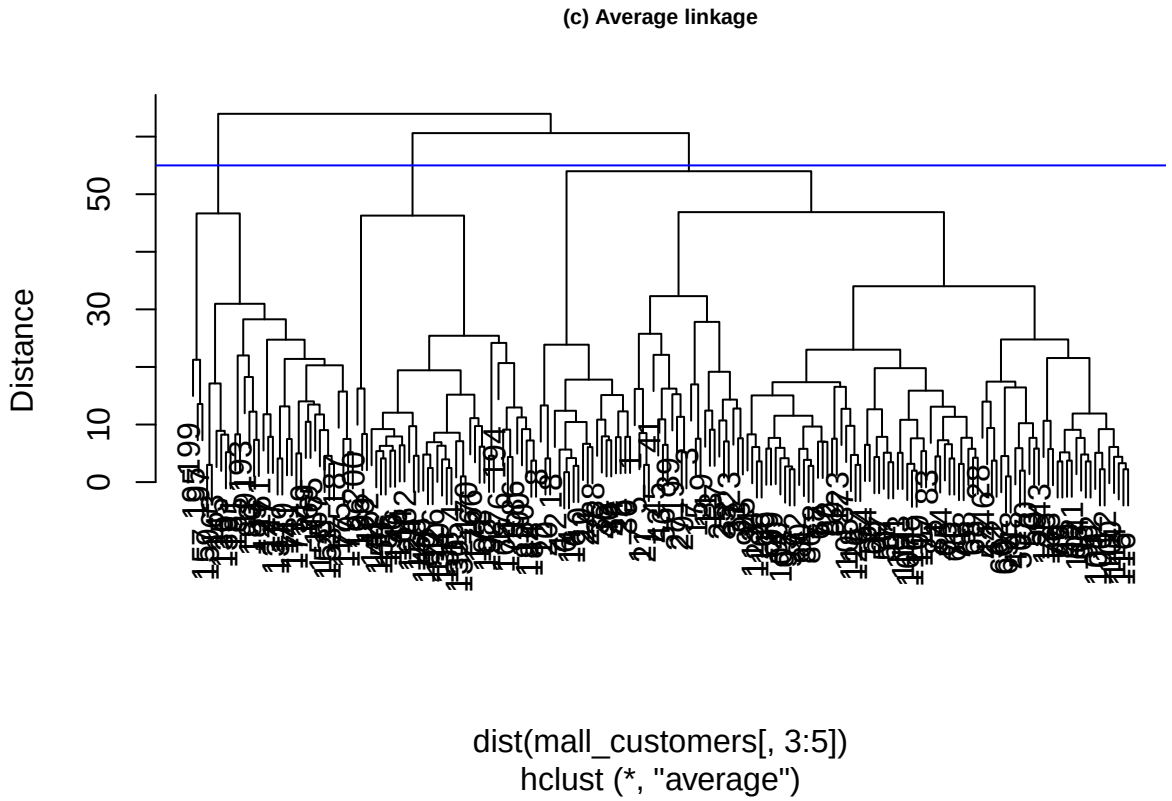
```
# Mean vectors for each cluster
cluster.mean2<-lapply(1:4,function(nc) {colMeans(mall_customers[cluster2==nc,3:5])})
cluster.mean2
```

```
## [[1]]
##           Age  AnnualIncome SpendingScore
##      42.82075      48.58491      43.50943
##
## [[2]]
##           Age  AnnualIncome SpendingScore
##         24.85         24.95         81.00
##
## [[3]]
##           Age  AnnualIncome SpendingScore
##      32.69231      86.53846      82.12821
##
## [[4]]
##           Age  AnnualIncome SpendingScore
##      41.68571      88.22857      17.28571
```

Hierarchical clustering using average linkage method

```
plot(hclust(dist(mall_customers[,3:5]),method="average"),
     labels=row.names(mall_customers[,3:5]),ylab="Distance",
```

```
     main="(c) Average linkage",cex.main=0.7)


#cutting the dendogram at height 800 to get various clusters formed at that point.
abline(h=55,col="blue")
```




```
cluster3<-cutree(hclust(dist(mall_customers[,3:5]),
                        method="average"),h=55)

max(cluster3)
```

```
## [1] 3
```

```
# Mean vectors for each cluster
cluster.mean3<-lapply(1:3,function(nc) {colMeans(mall_customers[cluster3==nc,3:5])})
cluster.mean3
```

```
## [[1]]
##           Age  AnnualIncome SpendingScore
##      39.96825      44.83333      49.46032
##
## [[2]]
##           Age  AnnualIncome SpendingScore
##      32.69231      86.53846      82.12821
##
## [[3]]
##           Age  AnnualIncome SpendingScore
##      41.68571      88.22857      17.28571
```

Hierarchical clustering with single linkage method: Cluster means:

Cluster no	Age	AnnualIncome	SpendingScore
1	38.81218	59.53807	50.64975
2	24.85	24.95	81.00
3	32	137	18

Hierarchical clustering with complete linkage method: Cluster means:

Cluster no	Age	AnnualIncome	SpendingScore
1	42.82075	48.58491	43.50943
2	24.85	24.95	81.00
3	32.69231	86.53846	82.12821
4	41.68571	88.22857	17.28571

Hierarchical clustering with average linkage method: Cluster means:

Cluster no	Age	AnnualIncome	SpendingScore
1	39.96825	44.83333	49.46032
2	32.69231	86.53846	82.12821
3	41.68571	88.22857	17.28571

Using single and average linkage methods I get 3 cluster and cluster means are different. But when I use complete linkage method I get 4 clusters and means of cluster 3 and 4 are similar to the means of cluster 2 and 3 from the method average linkage.

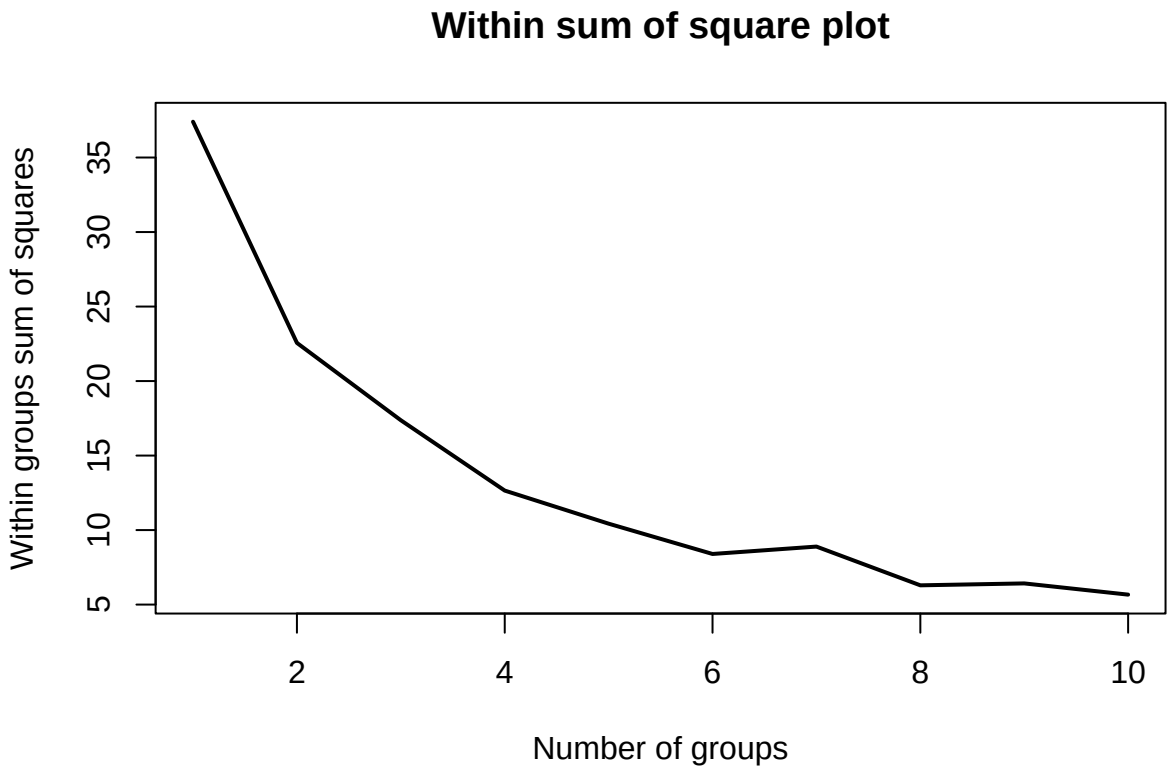
2. K-means clustering

Perform cluster analysis using K-means approach. The number of clusters are decide using the plot of within groups sum of squares.

```
#Standardizing the variables by dividing each variable by its range
#Finding min & max of each column (option 2) and doing max-min to get range
rge<-apply(mall_customers[,3:5],2,max)-apply(mall_customers[,3:5],2,min)

# Dividing entries of each column (option 2) by range
mall.std<-sweep(mall_customers[,3:5],2,rge,FUN="/")

set.seed(1234)
#Find sum of within-groups ss for #clusters = 1 to 10

wss<-numeric(10) #define wss as vector of length 10 (for #clusters = 1 to 10)
#within-group ss for two to six cluster solutions
for(i in 1:10) {
       W<-sum(kmeans(mall.std,i)$withinss)
       wss[i]<-W
}

wss
```

```
##  [1] 37.401482 22.560775 17.371946 12.650288 10.430324  8.398027  8.893652
##  [8]  6.293543  6.421693  5.668941
```

```
#Plotting the wss vs number of clusters
plot(1:10,wss,type="l",xlab="Number of groups",
     ylab="Within groups sum of squares",lwd=2,
     main="Within sum of square plot")
```




```
cat(" Plot suggest 4 or 6 clusters as optimum number of clusters")
```

```
##  Plot suggest 4 or 6 clusters as optimum number of clusters
```

```
# K-means output for K=4 clusters
mall.kmean<-kmeans(mall.std,4)
mall.kmean
```

```
## K-means clustering with 4 clusters of sizes 40, 57, 41, 62
##
## Cluster means:
##         Age AnnualIncome SpendingScore
## 1 0.6322115    0.7057377     0.8318878
## 2 0.4892038    0.3278689     0.6152882
## 3 0.7537523    0.6801279     0.2030861
## 4 1.0539702    0.3947647     0.4157340
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 3 2 4 2 4 2 4 2 4 2 2 2 4 2 2 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4
##  [38] 2 3 2 4 2 4 2 4 2 4 2 2 2 4 2 2 4 4 4 4 4 2 4 4 2 4 4 4 2 4 4 2 2 4 4 4 4
##  [75] 4 2 4 4 2 4 4 2 4 4 2 4 4 2 2 4 4 2 4 3 2 2 4 2 4 2 2 4 4 2 4 2 4 4 4 4 4
## [112] 2 3 2 2 2 4 4 4 4 2 3 1 1 3 1 3 1 4 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
## [149] 3 1 3 1 3 1 3 1 3 1 3 1 4 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3
## [186] 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
##
## Within cluster sum of squares by cluster:
## [1] 1.320230 3.747401 3.337115 4.252747
##  (between_SS / total_SS =  66.2 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
#Cluster means of unstandardized data saved in planet
lapply(1:4,function(nc) {apply(mall_customers[mall.kmean$cluster==nc,3:5],2,mean)})
```

```
## [[1]]
##           Age  AnnualIncome SpendingScore
##        32.875        86.100        81.525
##
## [[2]]
##           Age  AnnualIncome SpendingScore
##      25.43860      40.00000      60.29825
##
## [[3]]
##           Age  AnnualIncome SpendingScore
##      39.19512      82.97561      19.90244
##
## [[4]]
##           Age  AnnualIncome SpendingScore
##      54.80645      48.16129      40.74194
```

The wss plot suggest that 4 clusters are adequate. So used k-means cluster with 4 as the number of clusters.

Cluster means:

Cluster no	Age	AnnualIncome	SpendingScore
1	32.875	86.100	81.525
2	25.43860	40.00000	60.29825
3	39.19512	82.97561	19.90244
4	54.80645	48.16129	40.74194

3.Model-based clustering

Perform model-based clustering and make the pairwise scatterplots showing the clusters, density plot, and BIC plot.

```
library(mclust)
##without specifying the number of clusters
mb = Mclust(mall_customers[,3:5])
##number of clusters
mb$G
```

```
## [1] 4
```

```
##model name
mb$modelName
```

```
## [1] "VVI"
```

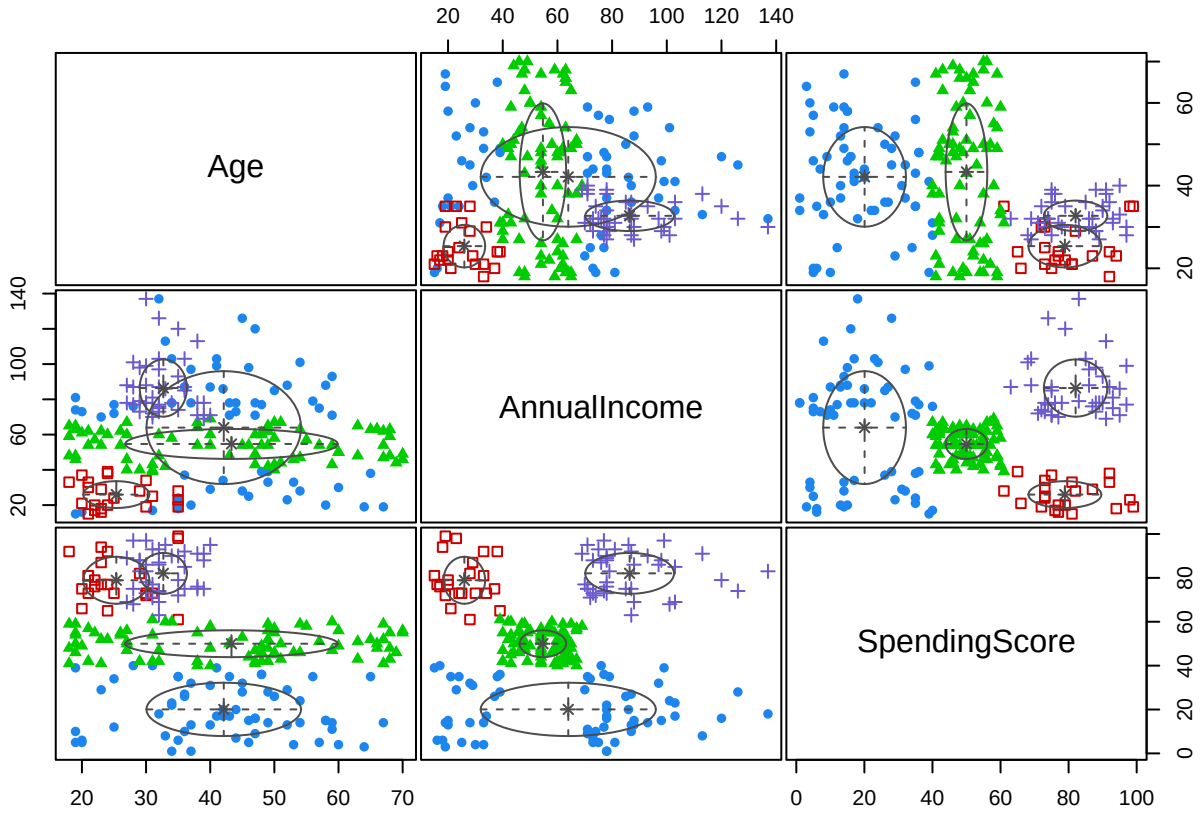
```
##cluster means

# get probabilities, means, variances
summary(mb, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVI (diagonal, varying volume and shape) model with 4 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2520.997 200 27 -5185.048 -5194.154
##
## Clustering table:
##  1  2  3  4
## 61 22 78 39
##
## Mixing probabilities:
##         1         2         3         4
## 0.3109140 0.1120534 0.3813049 0.1957277
##
## Means:
##                    [,1]     [,2]     [,3]     [,4]
## Age           42.11233 25.36187 43.31456 32.69209
## AnnualIncome  63.96406 25.95923 54.68602 86.40479
## SpendingScore 20.05959 78.90737 49.98842 82.05547
##
## Variances:
## [,,1]
##                     Age AnnualIncome SpendingScore
## Age           145.2871        0.000         0.000
## AnnualIncome    0.0000     1026.356         0.000
## SpendingScore   0.0000        0.000       147.768
## [,,2]
##                     Age AnnualIncome SpendingScore
## Age           26.50913      0.00000        0.0000
## AnnualIncome   0.00000     57.77577        0.0000
## SpendingScore  0.00000      0.00000      114.6515
## [,,3]
##                    Age AnnualIncome SpendingScore
## Age           274.434      0.00000       0.00000
## AnnualIncome    0.000     72.21181       0.00000
## SpendingScore   0.000      0.00000      37.46829
## [,,4]
##                     Age AnnualIncome SpendingScore
## Age           13.56566       0.0000       0.00000
## AnnualIncome   0.00000     263.7402       0.00000
## SpendingScore  0.00000       0.0000      86.92823
```

```
cat( "Pairwise scatterplots showing the clusters")
```

```
## Pairwise scatterplots showing the clusters
```

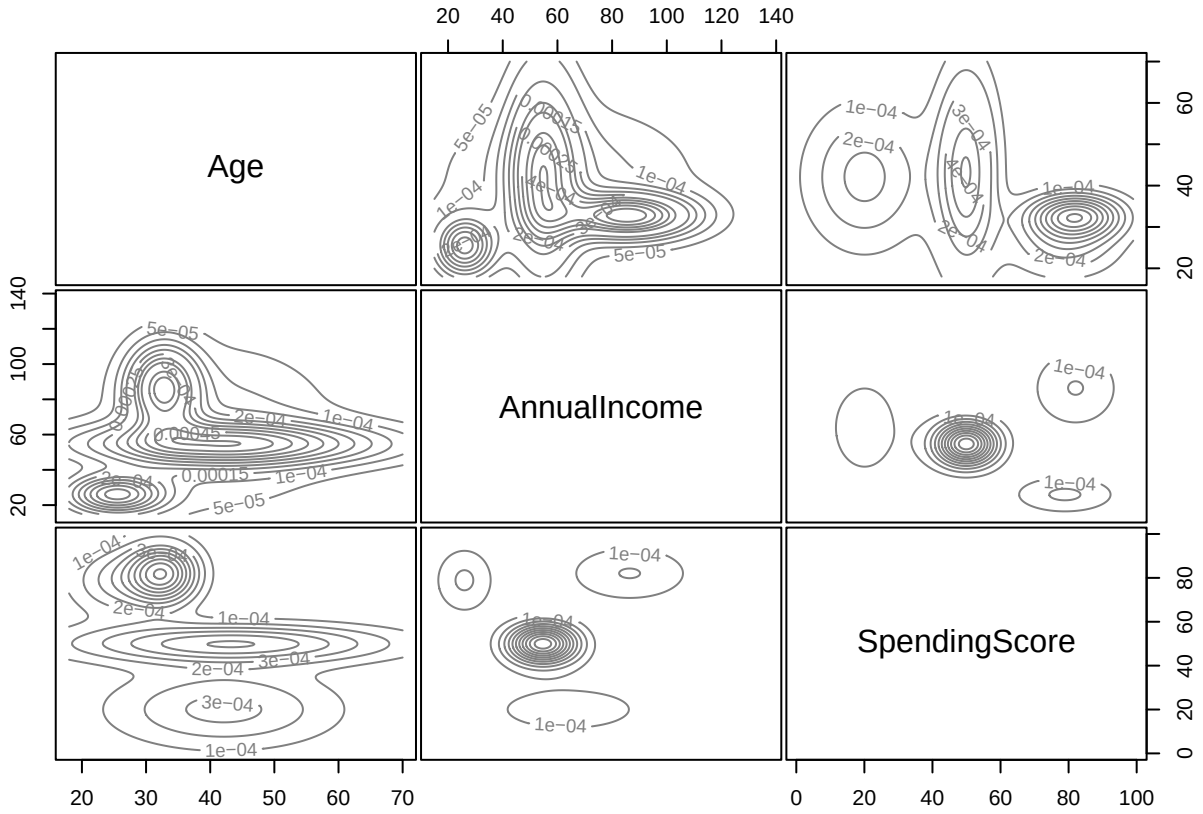
```
plot(mb, what=c("classification"))
```




```
cat("Density plot")
```

```
## Density plot
```

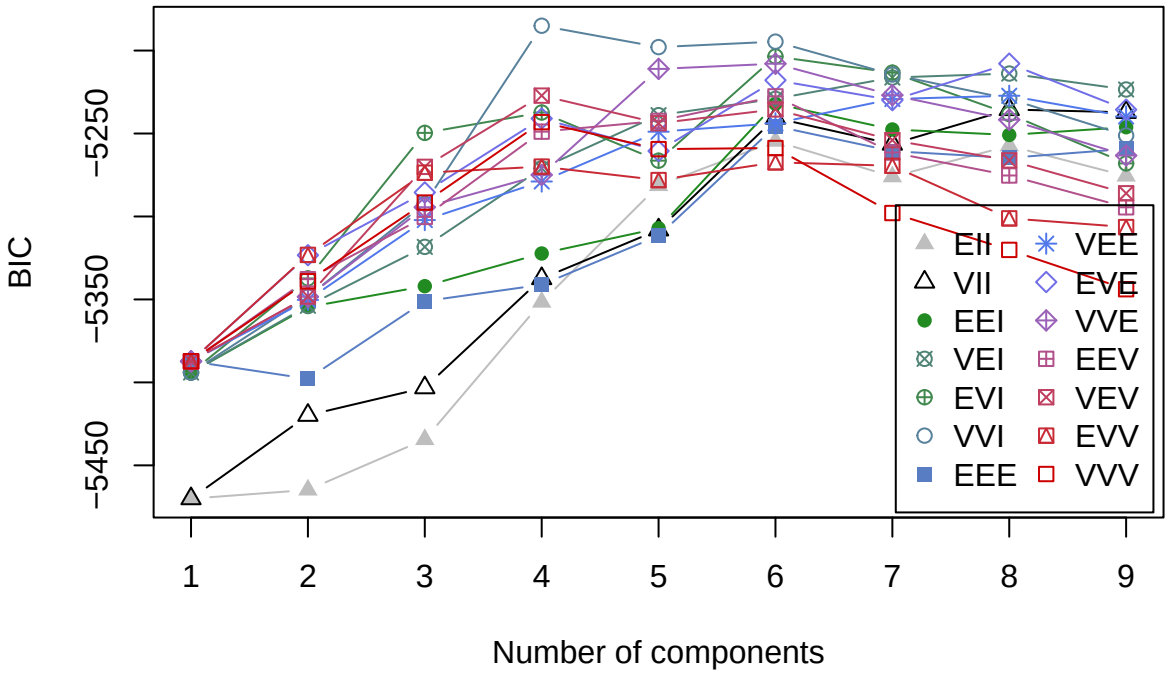
```
plot(mb, "density")
```




```
cat("BIC plot")
```

```
## BIC plot
```

```
plot(mb, "BIC")
```



Model based clustering

Model name :VVI

Number of optimum clusters : 4

Cluster means:

Cluster no	Age	AnnualIncome	SpendingScore
1	42.11233	63.96406	20.05959
2	25.36187	25.95923	78.90737
3	43.31456	54.68602	49.98842
4	32.69209	86.40479	82.05547

The pairwise scatterplots showing the clusters, density plot, and BIC plot were drawn.

Comparision the results from parts from above three methods. Also, use the package fpc to further compare the results.

```
library("fpc")

cs = cluster.stats(dist(mall_customers[,3:5]), mb$classification)
cs$within.cluster.ss #within cluster sum of squares
```

```
## [1] 128627
```

```
cs[c("within.cluster.ss","avg.silwidth")] #average silhouette width - ranges from 0 to 1; value closer to
```

```
## $within.cluster.ss
## [1] 128627
##
## $avg.silwidth
## [1] 0.3500368
```

-Using single and average linkage methods I get 3 cluster and cluster means are different. But when I use complete linkage method I get 4 clusters and means of cluster 3 and 4 are similar to the means of cluster 2 and 3 from the method average linkage.

-According to k-means cluster analysis and `Mclust` package output show that the optimal number of clusters is 4.

The `within cluster SS` of three clusters is 128627 by `fpc` output. It is a very high value. So less closely related objects are within the cluster.

Also the `average silhouette width` is 0.3500368«<1. This shows that the clusters are not clustered better.