Data_Mining_Assignment2_Duo_Zhou

Duo Zhou

7/11/2020

1. Select the numeric variables that you think are appropriate and useful. Use kmeans and Gaussian Mixture models. In this Boston Housing Data clustering analysis, numerical variables were chosen to apply Kmean and Gaussian Mixture on. I chose PTRATIO(pupil-teacher ratio), LSTAT(% lower status of the population) and MEDV (Median Housing Price) Clustering the first two variables can divide the cases into neighborhoods with different socio_economic status and educational resource levels. We can see how MEDV varies with different clustering of the first two variables.

```
# selecting variables PTRATIO and LSTAT
BH<-BH[,c('MEDV','LSTAT','PTRATIO')]
#seperating data into training and testing sets which contains 70% and 30% of total cases
# respectively
set.seed(12345)
BH_train <- sample_frac(BH,0.7)
BH_test <- setdiff(BH,BH_train)
dim(BH_train)
```

```
## [1] 354   3
```

```
dim(BH_test)
```

```
## [1] 151   3
```

```
BH_train_scale <- scale(BH_train)
BH_test_scale <- scale(BH_test)
BH_scale <-scale(BH)
```

2. Scale the data using standard scaling or 0-1 minmax scaling. R scale function does standard scaling.

3. Generate the K-means solution. Extract 2-10 k-means clusters using the variable set. Present the Variance-Accounted-For (VAF or R-square). Remember: the local optima problem is big for all the clustering and mixture models. So remember to run them from at least 50-100 random starts

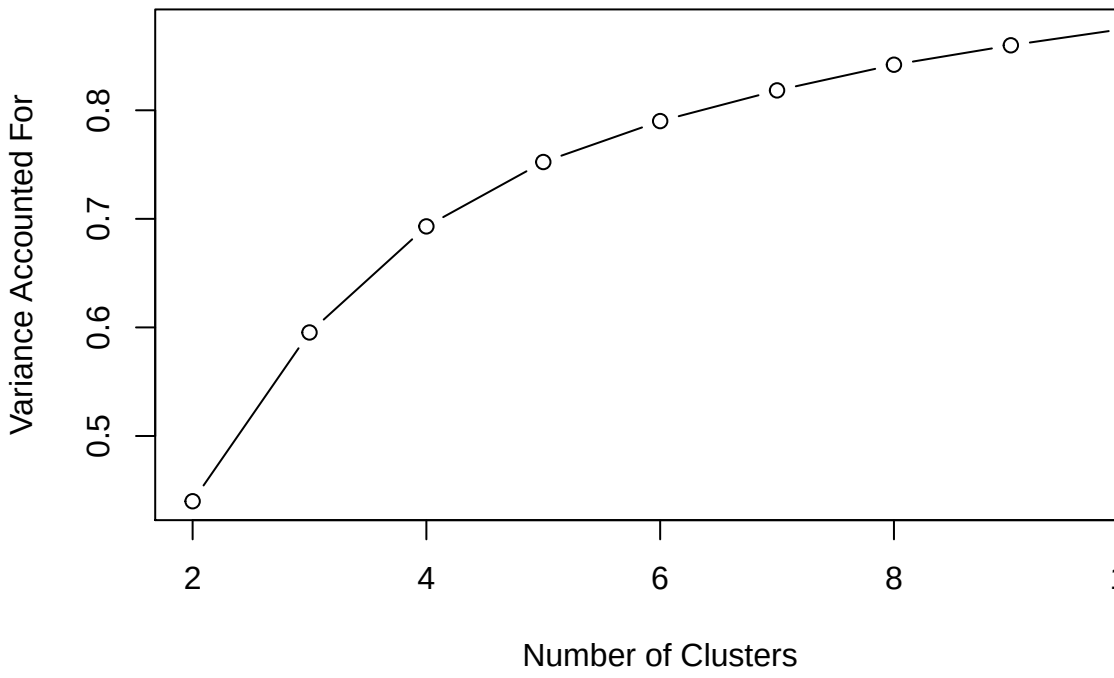
```r
set.seed(12345)
  vaf<-0
  for(i in 2:10){
       km<-kmeans(BH_train_scale,centers=i,nstart=100)
       vaf[i]<-km$betweenss/km$totss
}
## variance accounted for
kable(cbind(num.clusters=2:10,variance.accounted.for=vaf[-1]),caption="Variance Accounted for
by Number of Clusters Used")
```

4. Perform Scree tests to choose appropriate number of k-means clusters

Table 1: Variance Accounted for by Number of Clusters Used

num.clusters	variance.accounted.for
2	0.4398927
3	0.5953393
4	0.6930509
5	0.7523362
6	0.7900430
7	0.8183105
8	0.8419528
9	0.8598535
10	0.8754588

```r
## Scree Plot
plot(2:10,vaf[-1],type="b",xlab="Number of Clusters",ylab="Variance Accounted For")
```



5. Show the scree plot.

There is clearly an "elbow" at 3 clusters.

From The Scree plot above, we can see an distinct elbow exhibited at 3 clusters. Thus, we will proceed with using 3 clusters in our K-Means

```
# Running kmeans again using 3 clusters
K3<-kmeans(BH_train_scale,centers=3,nstart=100)
## number of observations in each cluster
kable(as.data.frame(table(K3$cluster)),col.names=c("Cluster","# of Obs."),caption="No. of
Observations per Cluster from Training dataset")
```

Table 2: No. of Observations per Cluster from Training dataset

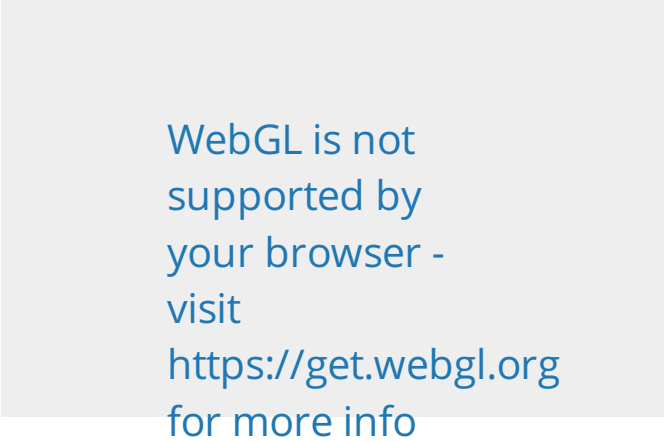
Cluster	# of Obs.
1	110
2	157
3	87

```
## Training VAF for 3 clusters
K3_VAF<-K3$betweenss/K3$totss
K3_VAF
```

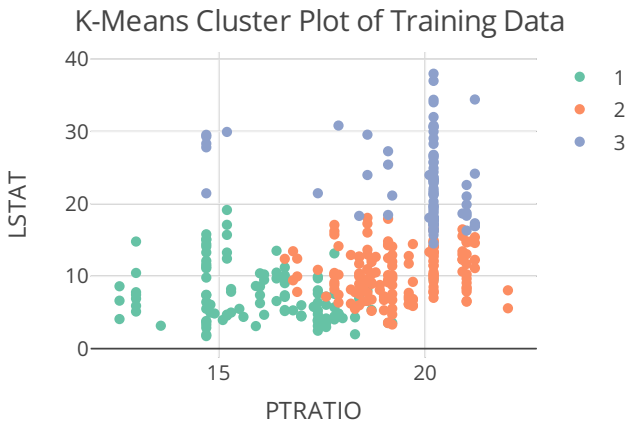
```
## [1] 0.5953393
```

```
## cluster plots of training data and centers of clusters. unscaled data

plot_ly(x=BH_train$PTRATIO,y=BH_train$LSTAT,z=BH_train$MEDV, type='scatter3d',mode='markers',
 color=as.factor(K3$cluster)) %>% layout(
           title='K-Means Cluster Plot of Training Data',
            scene=list(xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'),
                       zaxis=list(title='MEDV')))
```




```
plot_ly(x=BH_train$PTRATIO,y=BH_train$LSTAT, type='scatter',mode='markers',
 color=as.factor(K3$cluster)) %>% layout(
           title='K-Means Cluster Plot of Training Data',
            xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'))
```



In the above plots, we see each cluster representing different combinations of socio-economic status and education resource levels with Median Housing Price. Cluster 1 (green) mainly contain cases that are from towns with more education resources and good socio-economic status. AVG MEDV in cluster 1 is the highest. Cluster 2 (red) are cases that are from towns with less education resources but good socio-economic status. AVG MEDV in cluster 2 is in the middle. Cluster 3 (blue) are cases from towns wtih less education resources and worst socio-economic status. AVG MEDV in cluster 3 is in the lowest.

```
unscld_PTRA_mean<-tapply(BH_train$PTRATIO,K3$cluster,mean)
unscld_LLSTAT_mean<-tapply(BH_train$LSTAT,K3$cluster,mean)
unscld_MEDV_mean<-tapply(BH_train$MEDV,K3$cluster,mean)

# Table of centers of all clusters
kable(cbind(unscld_PTRA_mean,unscld_LLSTAT_mean,unscld_MEDV_mean),
      col.names=c("PTRATIO","LSTAT","MEDV"),
      row.names=TRUE,
      caption="Cluster Centers from Uncaled Training dataset")
```

Table 3: Cluster Centers from Uncaled Training dataset

	PTRATIO	LSTAT	MEDV
1	16.03909	7.38100	30.56727
2	19.35287	10.51172	21.34777
3	19.77931	22.66379	13.48391

This talbe containing the centers of the K-Means model of the training data set, shows the same information as the cluster plots.

6. Choose 1 K-means solution to retain from the many solutions that you have generated

a. Use the criteria of VAF.

b. Interpretability of the segments

```
K3_test<-kmeans(BH_test_scale,centers=K3$centers,nstart=100)
## number of obs in each cluster table(K3_test$cluster)
kable(as.data.frame(table(K3_test$cluster)),col.names=c("Cluster","# of Obs."),
caption="No. of Observations per Cluster from Test dataset")
```

c. Doing well in Test. For Test, use the centers (means) generated from the training set k-means solution, as the starting point for performing k-means in test. Use VAF and relative cluster sizes as measures of stability.

Table 4: No. of Observations per Cluster from Test dataset

Cluster	# of Obs.
1	34
2	63
3	54

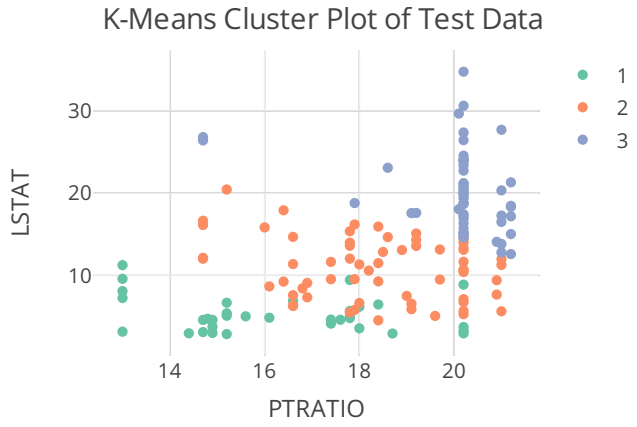
```
# Test VAF with 3 clusters
K3_test_VAF<-K3_test$betweenss/K3_test$totss
K3_test_VAF
```

```
## [1] 0.631937
```

```
## cluster plots of test data and centers of clusters. unscaled data
plot_ly(x=BH_test$PTRATIO,y=BH_test$LSTAT,z=BH_test$MEDV, type='scatter3d',mode='markers',
 color=as.factor(K3_test$cluster)) %>% layout(
           title='K-Means Cluster Plot of Test Data',
            scene=list(xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'),
                       zaxis=list(title='MEDV')))
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```
plot_ly(x=BH_test$PTRATIO,y=BH_test$LSTAT, type='scatter',mode='markers',
 color=as.factor(K3_test$cluster)) %>% layout(
           title='K-Means Cluster Plot of Test Data',
            xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'))
```




```
unscld_PTRA_mean_test<-tapply(BH_test$PTRATIO,K3_test$cluster,mean)
unscld_LLSTAT_mean_test<-tapply(BH_test$LSTAT,K3_test$cluster,mean)
unscld_MEDV_mean_test<-tapply(BH_test$MEDV,K3_test$cluster,mean)
## comparison of relative percentage of observations per cluster
kable(cbind(c("Train","Test"),rbind(table(K3$cluster)/length(BH_train[,1]),
table(K3_test$cluster)/length(BH_test[,1]))),
caption="Kmean Train and Test Relative Cluster Size")
```

Table 5: Kmean Train and Test Relative Cluster Size

	1	2	3
Train	0.310734463276836	0.443502824858757	0.245762711864407
Test	0.225165562913907	0.417218543046358	0.357615894039735

```
# comparison of centers of all clusters
kable(cbind(unscld_PTRA_mean_test,unscld_LLSTAT_mean_test,unscld_MEDV_mean_test,
            unscld_PTRA_mean,unscld_LLSTAT_mean,unscld_MEDV_mean),
      col.names=c("PTRATIO.Test","LSTAT.Test","MEDV.Test",
                  "PTRATIO.Train","LSTAT.Train","MEDV.Train"),row.names=TRUE,
      caption="Kmean Comparison of Centers from Uncaled Training and Test dataset")
```

Table 6: Kmean Comparison of Centers from Uncaled Training and Test dataset

	PTRATIO.Test	LSTAT.Test	MEDV.Test	PTRATIO.Train	LSTAT.Train	MEDV.Train
1	16.32059	5.27500	38.97941	16.03909	7.38100	30.56727
2	18.33016	10.85270	22.08254	19.35287	10.51172	21.34777
3	20.09444	20.13426	14.51667	19.77931	22.66379	13.48391

```
# comparison of VAFs between train and test
kable(cbind(K3_VAF,K3_test_VAF),col.names = c("Training VAF", "Test VAF"),
      caption="Kmean Comparison of VAFS bewteen Training and Test dataset")
```

Table 7: Kmean Comparison of VAFS bewteen Training and Test dataset

Training VAF	Test VAF
0.5953393	0.631937

We see that the the cluster centers as well as VAFs are very similar between taining and test data. The relative size of the cluster is little off with cluster 1 and cluster 3. We can not say that the our clustering is very stable, but it is good enough.

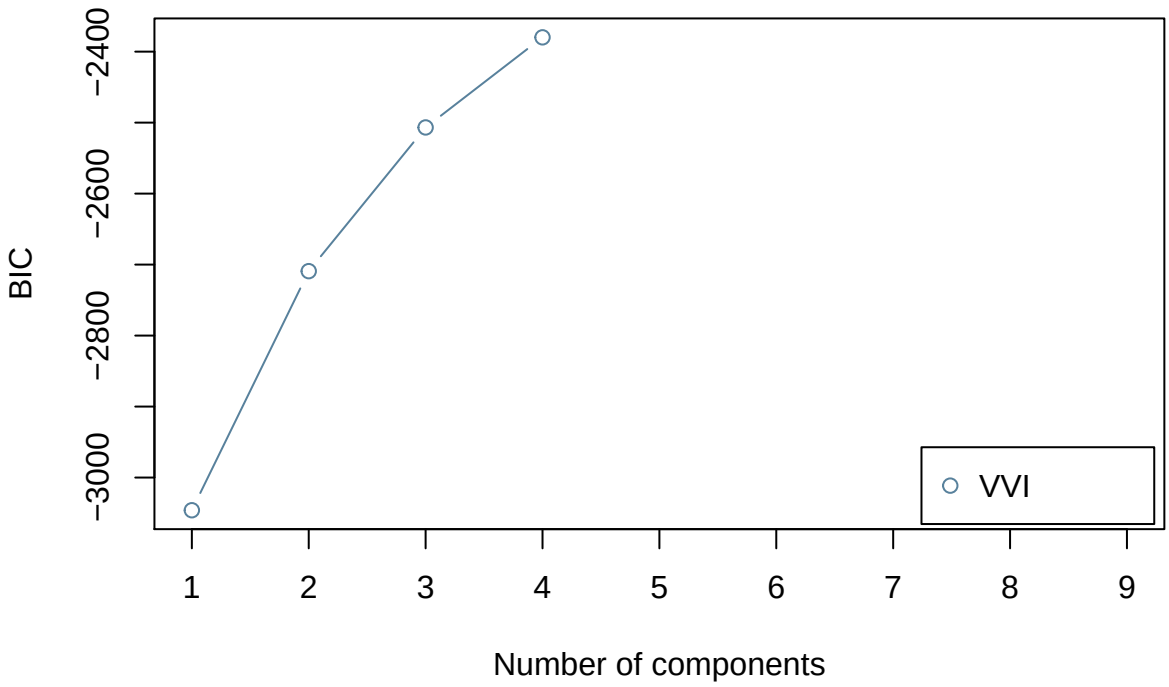
```
require(mclust)
```

7. Generate 3-5 Gaussian Mixtures (GM).

```
## Loading required package: mclust

## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

```
set.seed(12345)
BH.GM<-Mclust(BH_train_scale,modelNames = "VVI")
plot(BH.GM$BIC)
```




```
BH.GM$bic
```

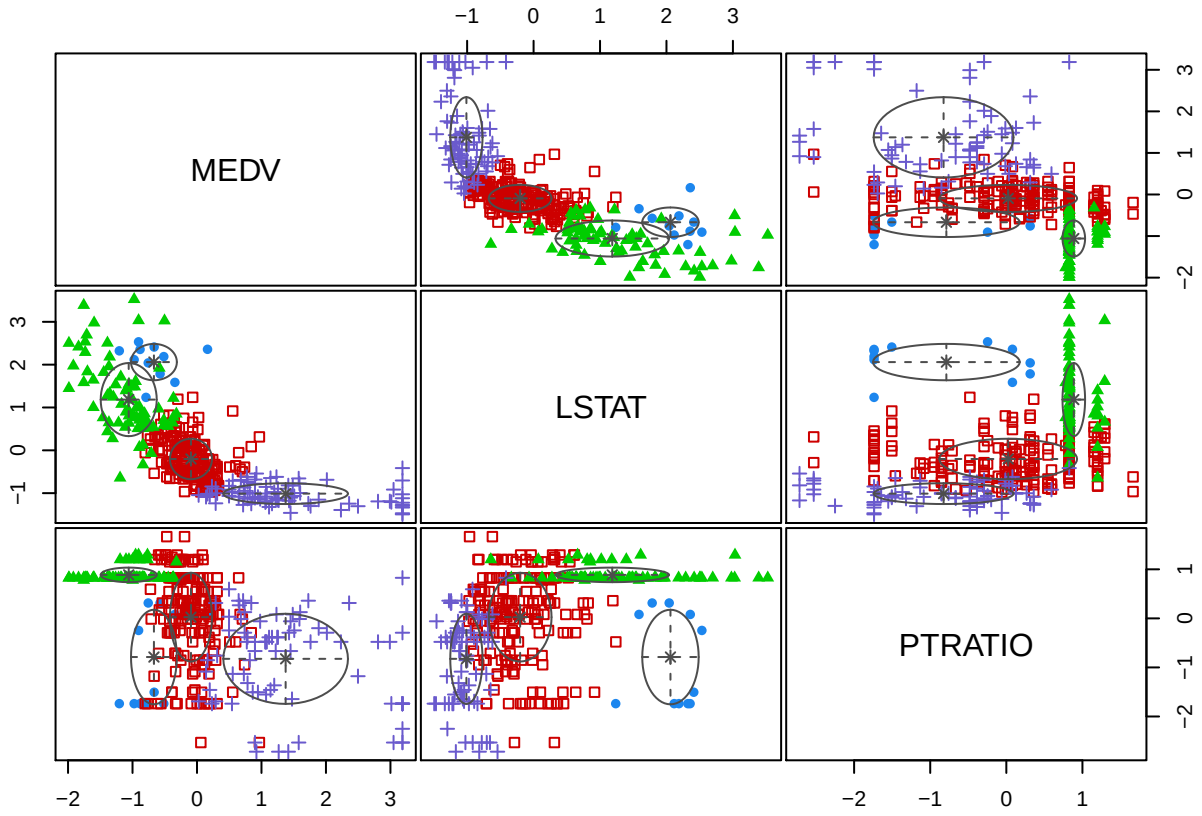
```
## [1] -2379.883
```

In this plot, we can see that 4 clusters produce the best (lowest absolute value)BIC. Our best GM model will use 4 clusters.

8. Compare the chosen k-means solution with the chosen GM solution from an interpretability perspective.

```
plot(BH.GM,what = 'classification',main="GM Cluster Plot of Scaled Training Data")
```

9. Summarize results and interpret the clusters/segments you choose as your final solution.

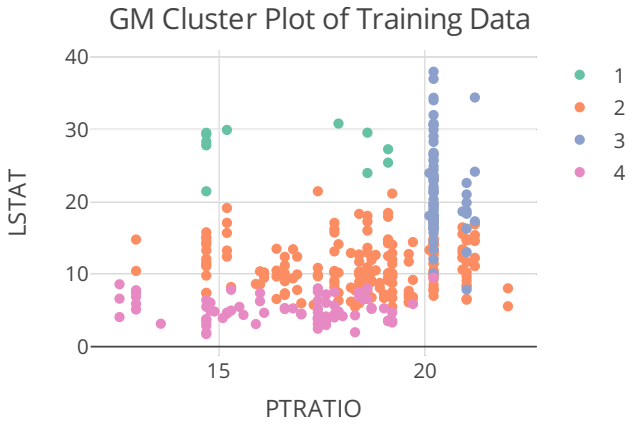



```
## cluster plots of training data and centers of clusters. unscaled data


plot_ly(x=BH_train$PTRATIO,y=BH_train$LSTAT,z=BH_train$MEDV, type='scatter3d',mode='markers',
 color=as.factor(BH.GM$classification)) %>% layout(
           title='GM Cluster Plot of Training Data',
            scene=list(xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'),
                       zaxis=list(title='MEDV')))
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```
plot_ly(x=BH_train$PTRATIO,y=BH_train$LSTAT, type='scatter',mode='markers',
 color=as.factor(BH.GM$classification)) %>% layout(
           title='GM Cluster Plot of Training Data',
            xaxis=list(title='PTRATIO'),yaxis=list(title='LSTAT'))
```




```
GM.unscld_PTRA_mean_train<-tapply(BH_train$PTRATIO,BH.GM$classification,mean)
GM.unscld_LLSTAT_mean_train<-tapply(BH_train$LSTAT,BH.GM$classification,mean)
GM.unscld_MEDV_mean_train<-tapply(BH_train$MEDV,BH.GM$classification,mean)
GM.unscld_PTRA_sd_train<-tapply(BH_train$PTRATIO,BH.GM$classification,sd)
GM.unscld_LLSTAT_sd_train<-tapply(BH_train$LSTAT,BH.GM$classification,sd)
GM.unscld_MEDV_sd_train<-tapply(BH_train$MEDV,BH.GM$classification,sd)

# comparison of means and sds of all clusters
kable(cbind(GM.unscld_PTRA_mean_train,
            GM.unscld_LLSTAT_mean_train,
            GM.unscld_MEDV_mean_train,
            GM.unscld_PTRA_sd_train,
            GM.unscld_LLSTAT_sd_train,
            GM.unscld_MEDV_sd_train),
      col.names=c("PTRATIO.Train.Mean",
                  "LSTAT.Train.Mean",
                  "MEDV.Train.Mean",
                  "PTRATIO.Train.sd",
                  "LSTAT.Train.sd",
                  "MEDV.Train.sd"
                  ),row.names=TRUE,
      caption="GM clusters means and sd of Uncaled Training dataset")
```

Table 8: GM clusters means and sd of Uncaled Training dataset

	PTRATIO.Train.Mean	LSTAT.Train.Mean	MEDV.Train.Mean	PTRATIO.Train.sd	LSTAT.Train.sd	MEDV.T
1	16.54545	27.575454	16.39091	2.0534671	2.872401	3.
2	18.51789	11.109474	21.43842	1.9303743	3.361320	2.
3	20.33026	21.424474	12.76974	0.3128449	6.123324	3.
4	16.59740	5.085844	34.58442	1.9746401	1.677614	8.

```
kable(table(BH.GM$classification),col.names=c("Clusters",'Number of Obs.'),
      caption="Number of Obs. in Each Cluster")
```

Table 9: Number of Obs. in Each Cluster

Clusters	Number of Obs.
1	11
2	190
3	76
4	77

From the above graph one can see that, best GM models gives 4 clusters and the best Kmean model gives only three. Based on the clustering plots of unscaled data, the GM model clusters contain much fewer outliers than the kmean clusters. That is because GM models take into the consideration of each observation's Gaussian probability densites with different means and sds. On the other hand, kmean models only cares about how close is each observation to the cluster means. The final model selection is GM Model with 4 clusters. Cluster 1(green) of the final model contains mainly the cases from towns with good education resources but the lowest socio-economic status. The AVG MEDV of cluster 1 is the second lowest. Cluster 2(Red) of the final model contains mainly the cases from towns with less education resources but hight socio-economic status. The AVG MEDV of cluster 2 is the second highest. Cluster 3(blue) of the final model contains maily the cases from towns with lower socio-economic status and least education resources. The AVG MEDV of cluster 2 is the lowest. Cluster 4(Purple) of the final model contains maily the cases from towns with the highest socio-economic status and very good education resources. The AVG MEDV of cluster 4 is the highest.

10. You are given the task of recruiting 30 people (per segment) into these segments for focus groups and other follow-up A&U (Attitudinal and Usage studies). a. What approach will you take to recruit people over the telephone? b. Assume consumers who are recruited will be reimbursed for either coming to a focus group or for AUU surveys. Which of the consumers will you try to recruit? c. How will you identify if a new recruit belongs to a particular segment. From our segmentation, these cluster 3 and 4 contain lowest and highest Median Housing Prices repectively. The average education resouces and socioeconomic status of these two clusters are also at two opposite end of the spetrum. They are of the highest interst to us for they can give a clear insight the effect of local education and economic status on housing prices. For the purpose of creating focus groups for use in Attitudinal and Usage studies, we would like to focus on house owners that fall within these segments.

For recruiting people into such a A&U study focus groups, we can identify potential candidates first by choosing people whose houses fall within those ranges. For overlapping situations, we can caculate the pnorm of candidate using mean and sd from each of the two segments. Whichever Gaussian distribution generates higher probability, the cadidate will fall in the that perticular cluster.

We could randomly sample 30 per each segment using above creteria and call them to request their participation in the study. For any number of candidates that decline participation, we would then randomly

resample the remainder of candidates and continue calling, repeating this process until we reach the required 30 people per segment.