SAE-Part3

Created by John Lawler - JML190001 - Team 7

CS4375.004 with Karen Mazidi

Created on 3/18/2023, last worked on 3/25/2023

The website given in the PDF outline is here (https://www.statmethods.net/advstats/cluster.html). This has a lot of the information for creating all of the three cluster types talked about in this document as well as some other information relevant to clustering. Additionally, this (https://stats.stackexchange.com/questions/263374/clusters-and-data-visualisation-in-r) website was most helpful for creating the more readable plot for kMeansClusters and Model Clusters.

kMeans Cluster

Here, we are getting everything ready to perform the kMeans cluster by reading in the csv file (that is in the same folder as this .Rmd file), setting a seed for reproducability, and scaling our numeric vectors we want to look at (in this case: PRICE, BEDROOMS, BATHROOMS).

```
perth <- read.csv("perth.csv")
perthScale <- scale(perth[3:5]) # scale our numbers
set.seed(1234) # reproducability
```

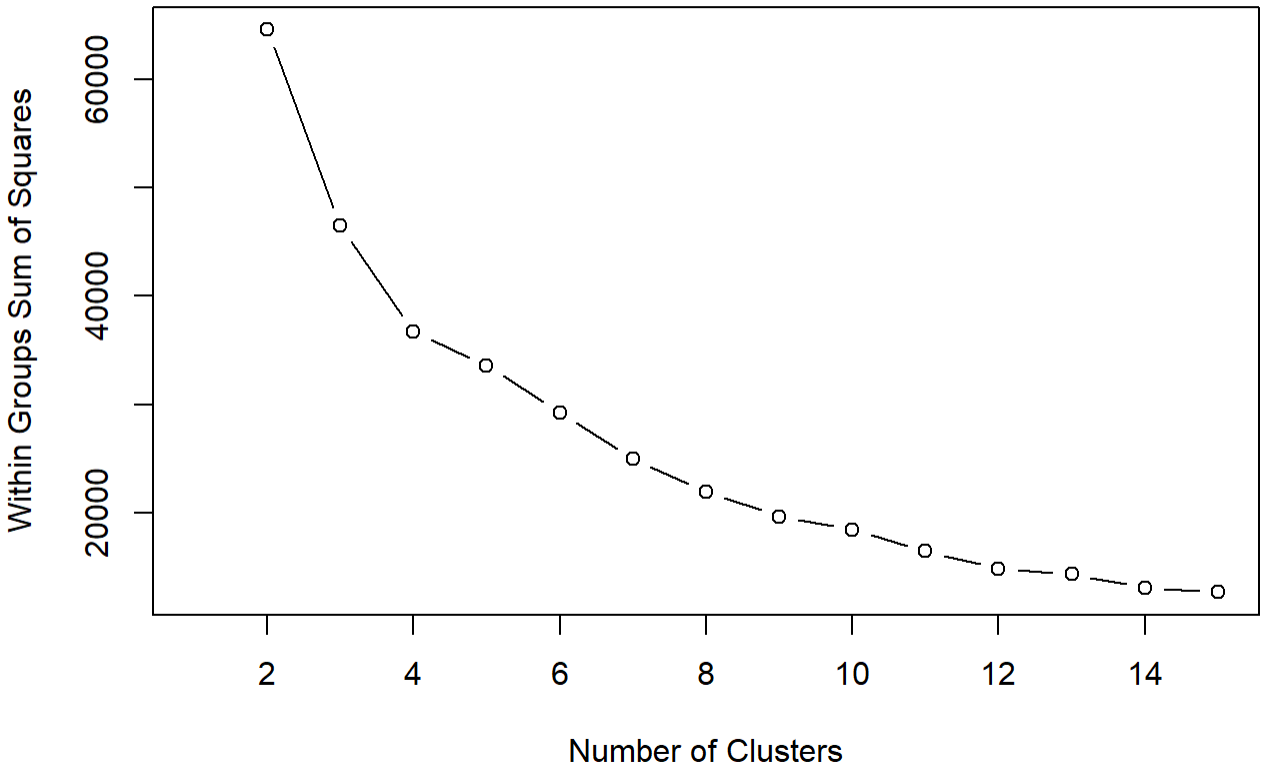
Finding the Number of Clusters

Here, we are looking for the optimal number of clusters. In this case, 8 seems like a good option as that is where it begins to taper off, but 12 is also a viable option as it is closer to the bottom and any additional number of clusters provides little improvement beyond that. In this case, I will use 12 because it does not add very much time to program, and we will see better fitting results.

```
wss <- (nrow(perth)-1) * sum(apply(perth, 2, var))
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```
for (i in 2:15) wss[i] <- sum(kmeans(perthScale, centers = i)$withinss)
plot(1:15, wss, type = "b", xlab = "Number of Clusters", ylab="Within Groups Sum of Squares")
```



Perform the kMeans Cluster Perform the cluster with 8 and 12 clusters to compare results. Doing 4 clusters as well to show POV statistic.

```
kMeansFour <- kmeans(perthScale, centers = 4, nstart = 20)
kMeansEight <- kmeans(perthScale, centers = 8, nstart = 20)
kMeansResult <- kmeans(perthScale, centers = 12, nstart = 20)
```

Summary of kMeansResult

Here is the raw data of the cluster. We will look at the 12 cluster model as it is more accurate without being an inordinate amount of clusters for the data given (about 33,000 observations) The most important parts to look at here are the cluster means and the “within cluster sum of squares by cluster” percentage, showing us we get above 50% on the percentage, and that our values in the cluster means are all within [-1,1] for the most part with a handful of exceptions reaching about [-1.4, 2.96] as there will be outliers in home price. At this point, we append the clusters to our data for interpretation. While seeing the kMeansResult data, it will show all 30,000 assignments in the knit file, so I have commented it out. In the next chunk, we will take a look at the most relevant data from it, showing the POV.

```
summary(kMeansResult)
```

```
##              Length Class  Mode
## cluster      33656  -none- numeric
## centers         36  -none- numeric
## totss            1  -none- numeric
## withinss        12  -none- numeric
## tot.withinss     1  -none- numeric
## betweenss        1  -none- numeric
## size            12  -none- numeric
## iter             1  -none- numeric
## ifault           1  -none- numeric
```

```
#kMeansResult
perthScale <- data.frame(perthScale, kMeansResult$cluster) # add the clusters to our data
```

Percentage of Varariance

This is the POV statistic for kMeans clusters with clusters of 4, 8, and 12. While 4 is pretty unoptimal, it is a “normal” value for clusters in kMeans clustering, and I wanted to show it’s value on this dataset. Additionally, I tried with a recommended sqrt of the length of the data, but this creates somewhere around 180 clusters with a POE of about 96% which introduces a level complexity and overfitting that is ultimately unhelpful. Again we will be sticking to 12 clusters as the percentage is 85.66% which is very good and will be good for our model.

```
((kMeansFour$totss - kMeansFour$tot.withinss) / kMeansEight$totss) * 100  # 4 cluster POV
```

```
## [1] 63.63135
```

```
((kMeansEight$totss - kMeansEight$tot.withinss) / kMeansEight$totss) * 100 # 8 cluster POV
```

```
## [1] 78.5397
```

```
((kMeansResult$totss - kMeansResult$tot.withinss) / kMeansResult$totss) * 100 # 12 cluster PO
V
```

```
## [1] 85.66493
```

Required Packages

In order to create the plot, you will need these packages. Additionally, you will need rlang 1.1.0 and vctrs 0.6.0. I had to do this manually, and it is a pain to do, but you can look at the plot after the next code chunk and save yourself the trouble. Additionally, I think you need devtools and another thing to install, so I do not recommend running this yourself if you want to save 30mins - 1hr of trying to install new libraries.

```
if(!require(factoextra)) install.packages("factoextra")
```

```
## Loading required package: factoextra
```

```
## Loading required package: ggplot2
```

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

```
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(devtools)) install.packages("devtools")
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

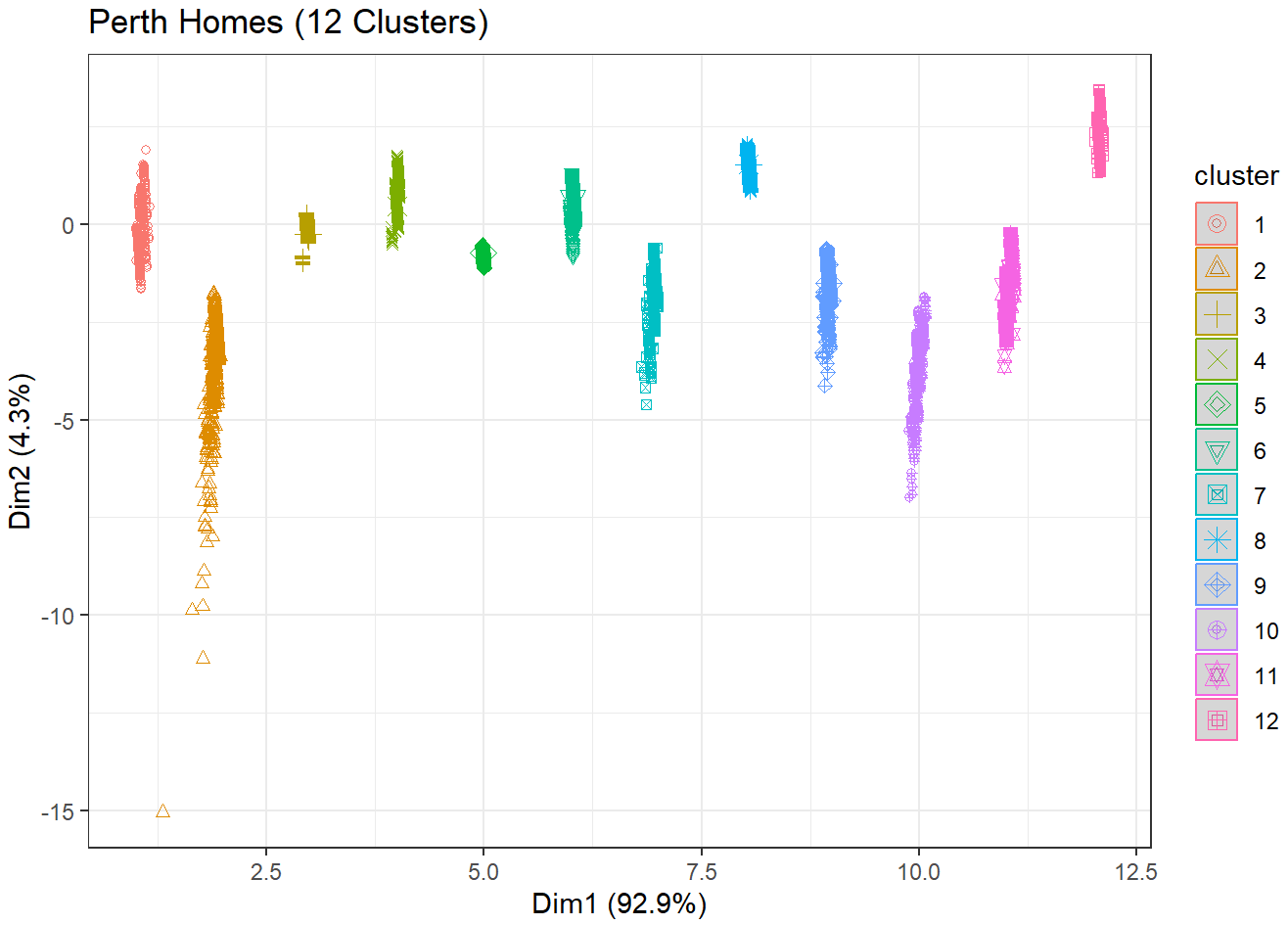
```
if(!require(fpc)) install.packages("fpc")
```

```
## Loading required package: fpc
```

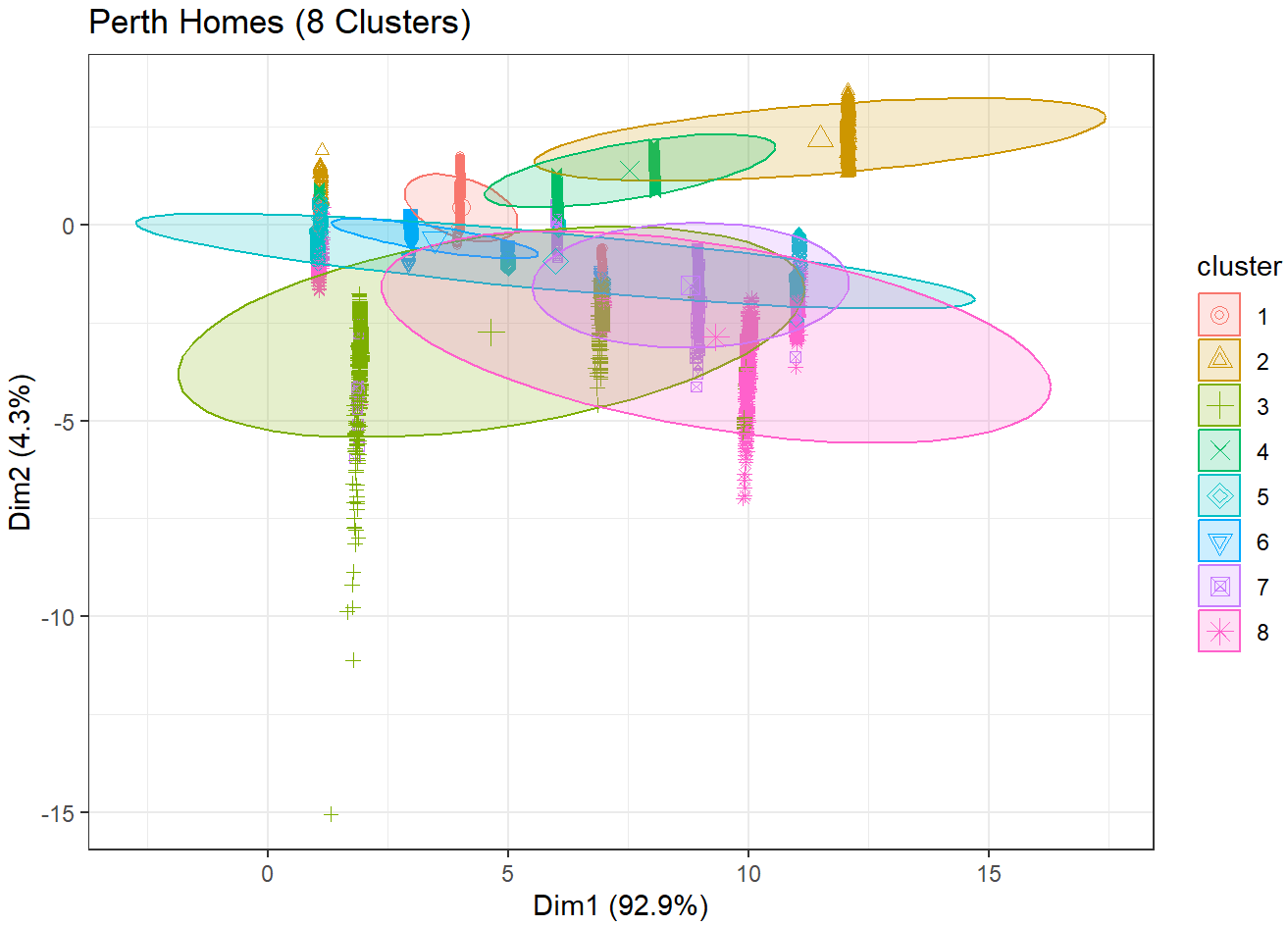
Graph of kMeans Cluster

In summary what we can tell from this is that, with about 85.66% accuracy, homes in Perth can be clustered into 12 groups. A lower number of groups creates overlap (which can be explained by cheaper homes having similar features of 1 bed, 1 bath, or something akin to that). The percentages along the axes corresponds to how much variance that dimension causes which can probably be attributed to factors outside BATHROOMS and BEDROOMS having an effect on the price of the home (such as location, the year it was built, any damage the home has endured, garages, etc.). I find the second graph to be rather unhelpful as there are too many data points and overlap to really understand what is going on, but as a whole we can take away that factors other than the ones we used in perthScale have an effect on the home price. I additionally showed the graph with 8 clusters as the overlap shows us a better idea of these outside factors.

```
library(cluster)
library(devtools)
library(ggplot2)
library(factoextra)
library(NbClust)
options(warn=-1)
# 12 cluster
fviz_cluster(kMeansResult, data = perthScale, geom = "point", stand = FALSE, frame.type = "no
rm", main = "Perth Homes (12 Clusters)") + theme_bw()
```

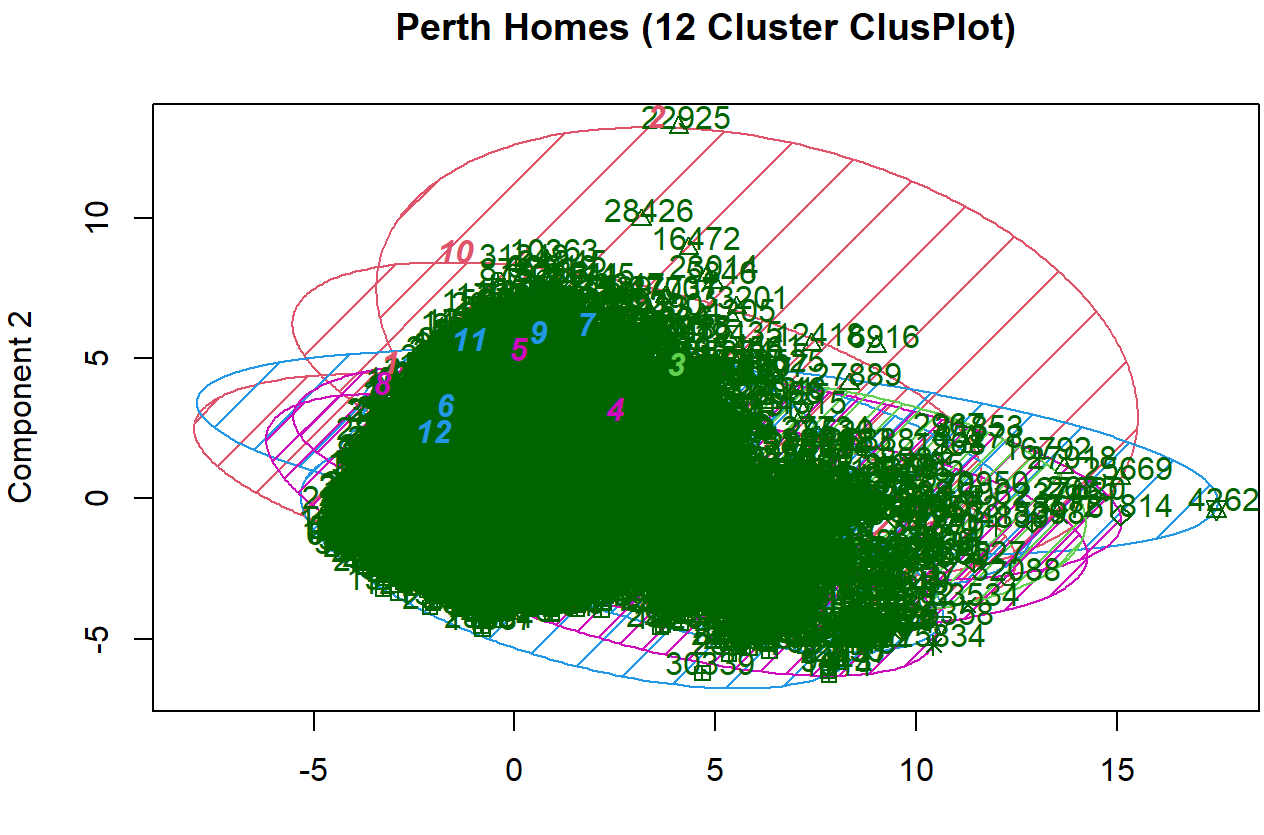



```
# 8 cluster
fviz_cluster(kMeansEight, data = perthScale, geom = "point", stand = FALSE, frame.type = "nor
m", main = "Perth Homes (8 Clusters)") + theme_bw()
```




```
options(warn=1)
# additional 12 cluster graph
clusplot(perth, kMeansResult$cluster, color=TRUE, shade=TRUE, labels=2, lines=0, main = "Pert
h Homes (12 Cluster ClusPlot)")
```

```
## Missing values were displaced by the median of the corresponding variable(s)
```



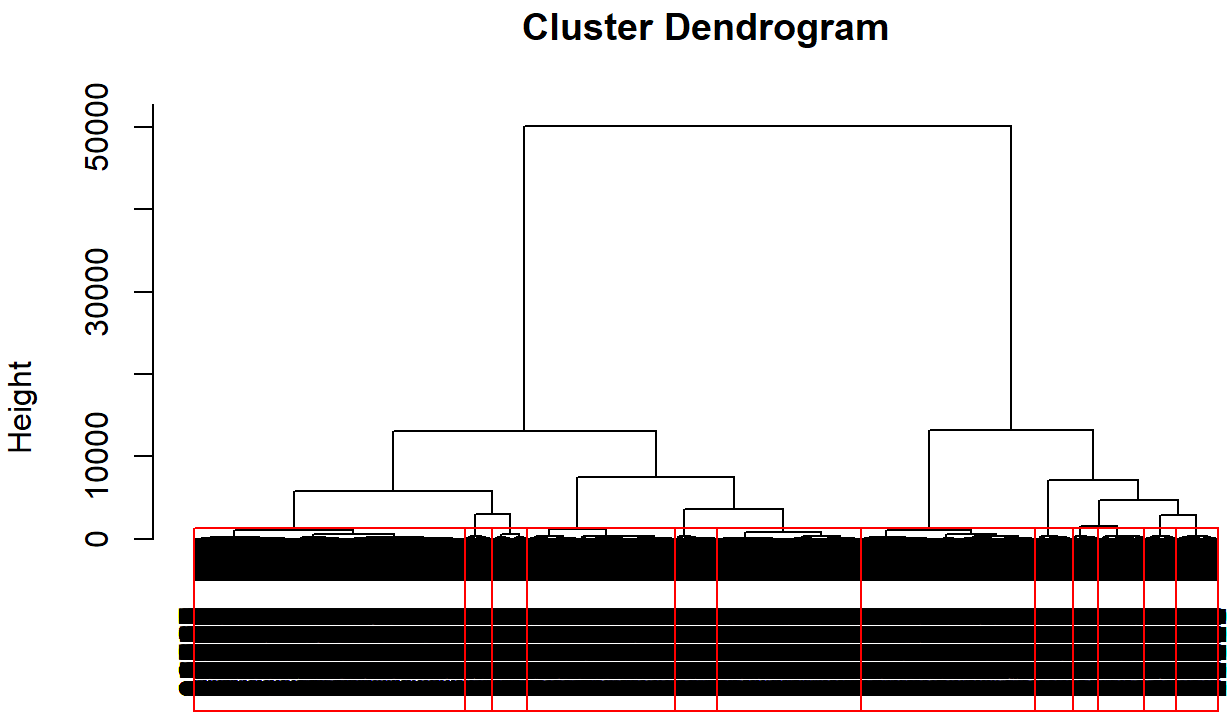
Hierarchal Cluster We already have all of the setup to do a Hierarchical Cluster, so go ahead and make a distance matrix with the euclidean method. Please note that this will take a massive amount of RAM, as this one matrix is roughly 4.5 gigabytes. From here on, I would not recommend running this with anything less than 16 GB of RAM and I would recommend that you have 32 GB or clearing earlier chunks. Or, you could create a subset to use less data.

```
distMatrix <- dist(perthScale, method = "euclidean") # create matrix for hierarchal cluster
```

The Cluster

The website given uses ward method (note that the ward method has been changed to ward.D) to make the hierarchical cluster. We then plot this dendrogram and cut our tree into the earlier determined 12 clusters, giving them red borders to more easily see them. There are a few outliers, but for the most part, all of the prices can be clustered into 12 groups. It also shows us the distribution of the costs in the houses, showing density among the houses according to their cluster prices.

```
fit <- hclust(distMatrix, method = "ward.D")
plot(fit) # plot the graph
groups <- cutree(fit, k=12) # cut our clusters
rect.hclust(fit, k=12, border="red") # highlight clusters
```



PValue Cluster Dendrogram This chunk can take some time to execute. This can show us whether the clusters are statistically significant, providing evidence for differences between groups. In this case, the graph is a bit difficult to read. The height between the kMeansResult cluster is pretty far off from the PRICE BEDROOMS and BATHROOMS branches which in this case, I believe means that it is a very low p-value. This means that our result indicates statistical significance between these factors.

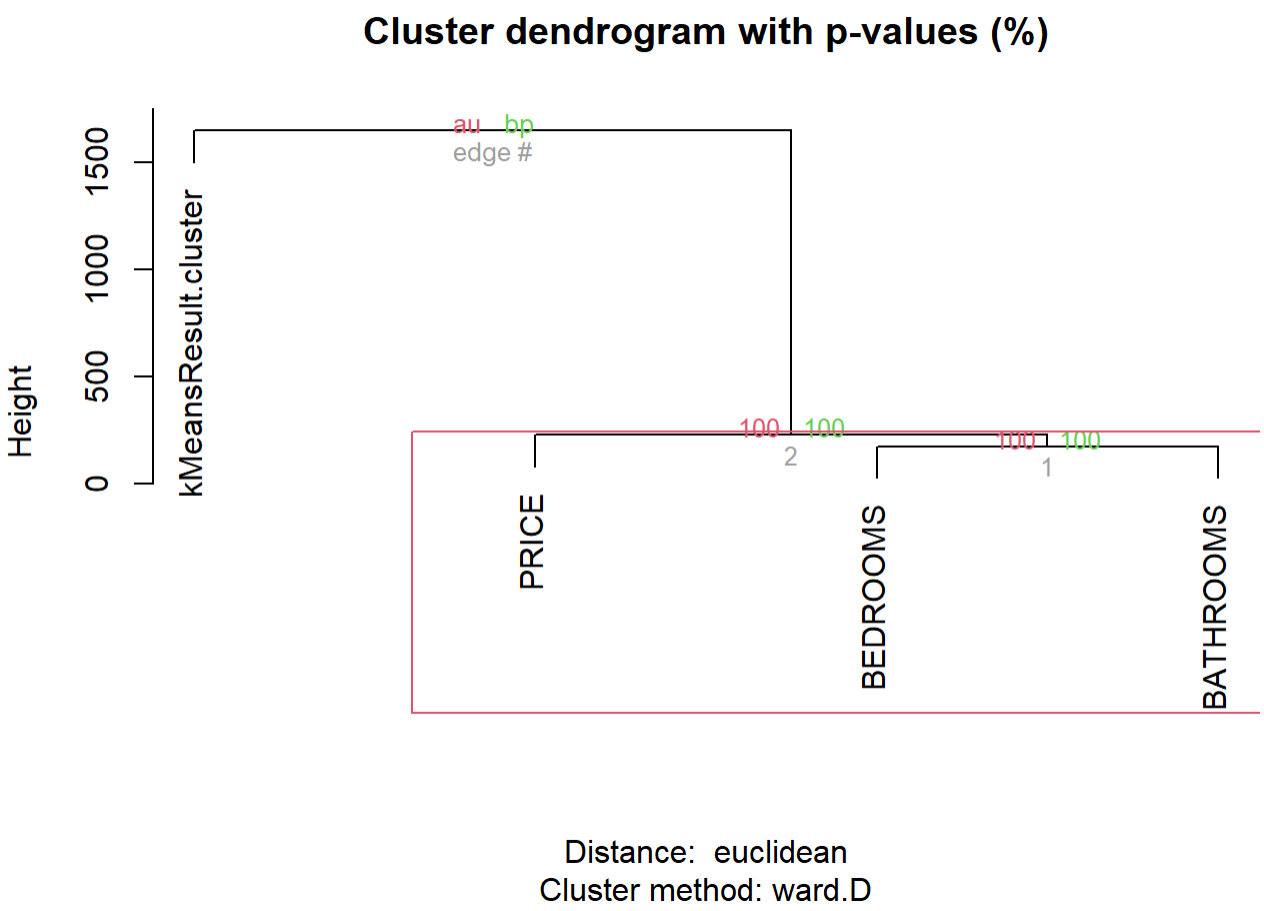
```
# required library
if(!require(pvclust)) install.packages("pvclust")
```

```
## Loading required package: pvclust
```

```
library(pvclust)

fit <- pvclust(perthScale, method.hclust="ward.D", method.dist="euclidean") # pvalue cluster dendrogram
```

```
## Bootstrap (r = 0.5)... Done.
## Bootstrap (r = 0.6)... Done.
## Bootstrap (r = 0.7)... Done.
## Bootstrap (r = 0.8)... Done.
## Bootstrap (r = 0.9)... Done.
## Bootstrap (r = 1.0)... Done.
## Bootstrap (r = 1.1)... Done.
## Bootstrap (r = 1.2)... Done.
## Bootstrap (r = 1.3)... Done.
## Bootstrap (r = 1.4)... Done.
```

```
plot(fit) # plot the pvalue cluster dendrogram
pvrect(fit, alpha=.95)
```



Model Based Clustering First we need the model clustering library.

```
if(!require(mclust)) install.packages("mclust")
```

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

```
## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(mclust)
```

Creating the Model Cluster

Create the cluster model on our scaled data. This one has a loading bar for its progress.

```
modelClust <- Mclust(perthScale)
```

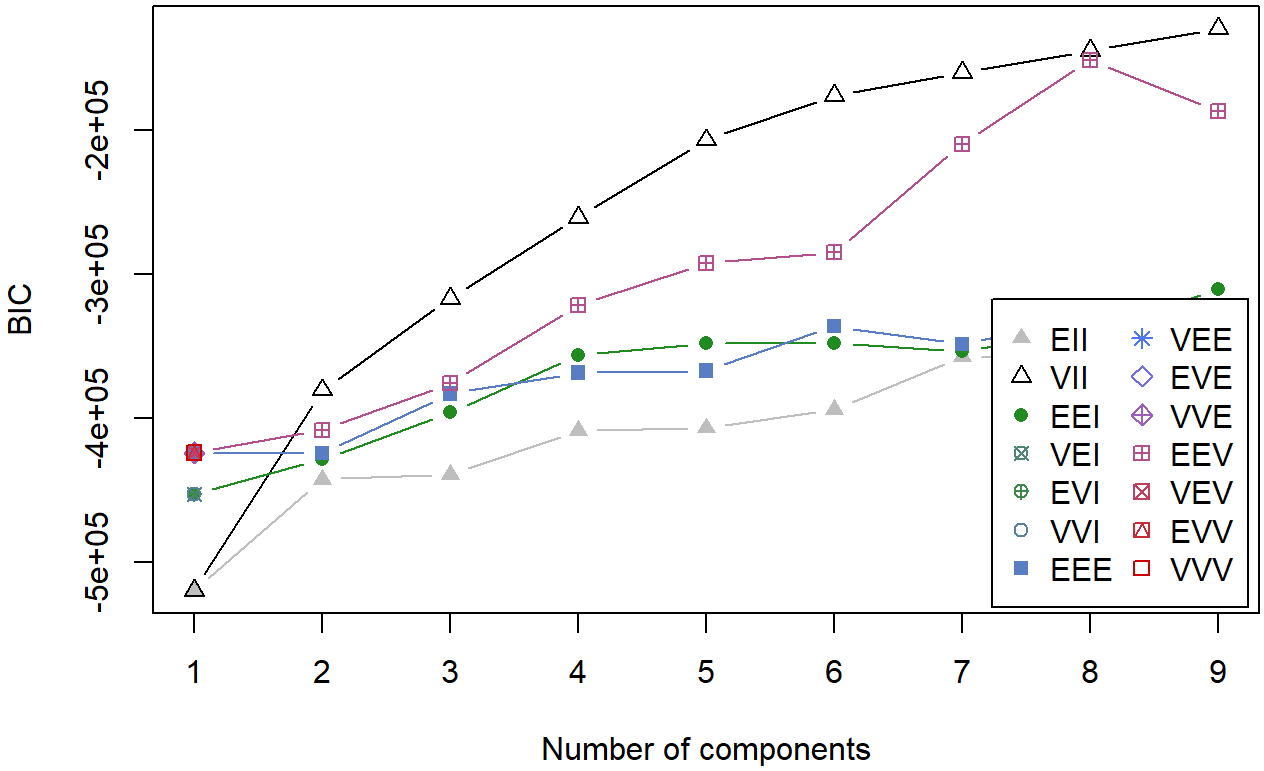
Plotting the Model Cluster

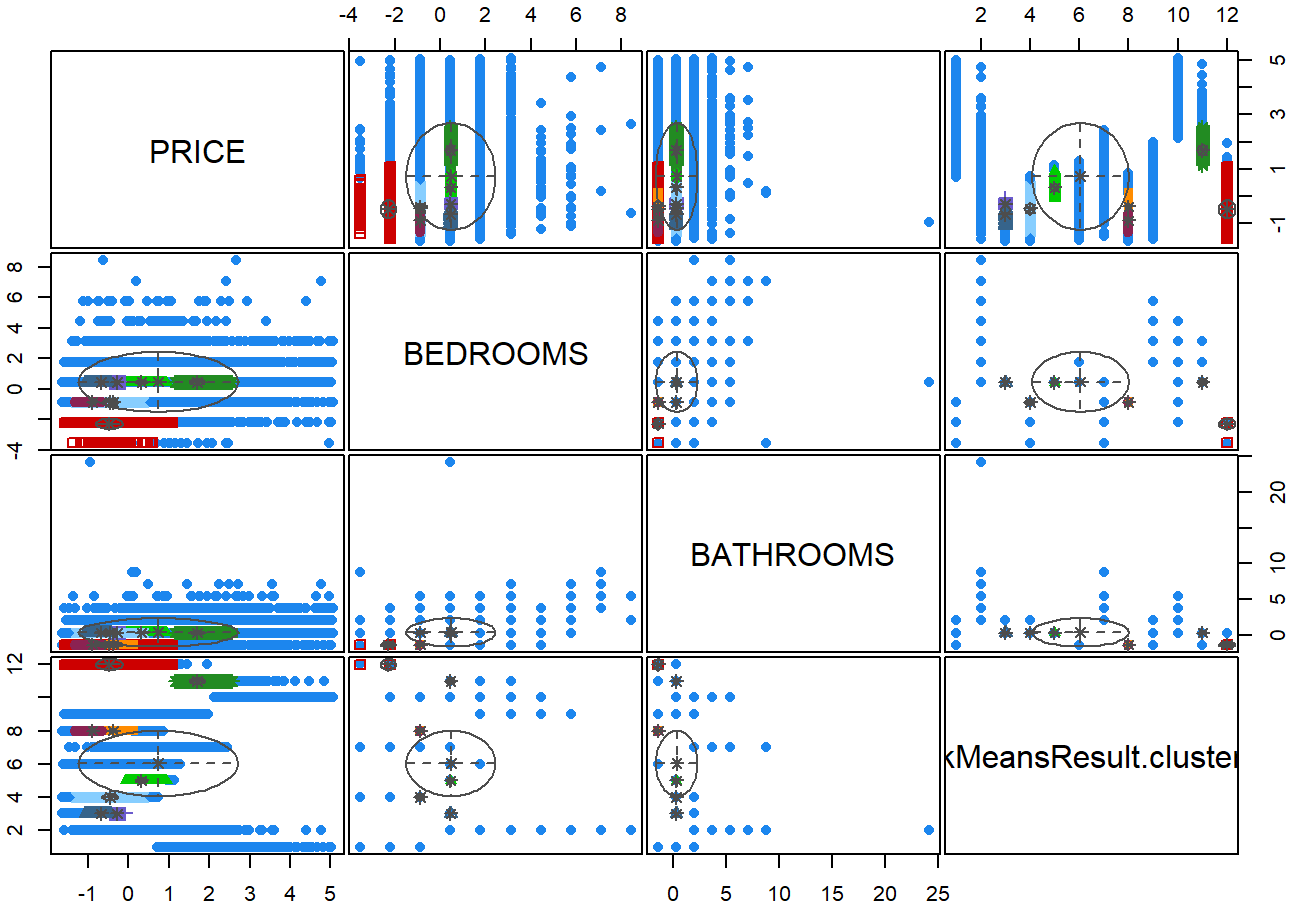
This one is the simplest to implement, but it also does take some time to produce. You will need to use the console and enter all 4 selections to see the graphs and then enter a '0' to close it out; however, in the document, they should all appear. While all four are helpful, we will mainly talk about the classification graph. The specific part we will take a look at as its difficult to read are the ellipses. These show the covariance for the variables, and judging by how they are shaped, We can see a definite relation (according to covariance) between PRICE BEDROOMS and BATHROOMS. There is a lot of data that lies outside the ellipses (in the form of the many split lines), but the clusters remain in their ellipses.

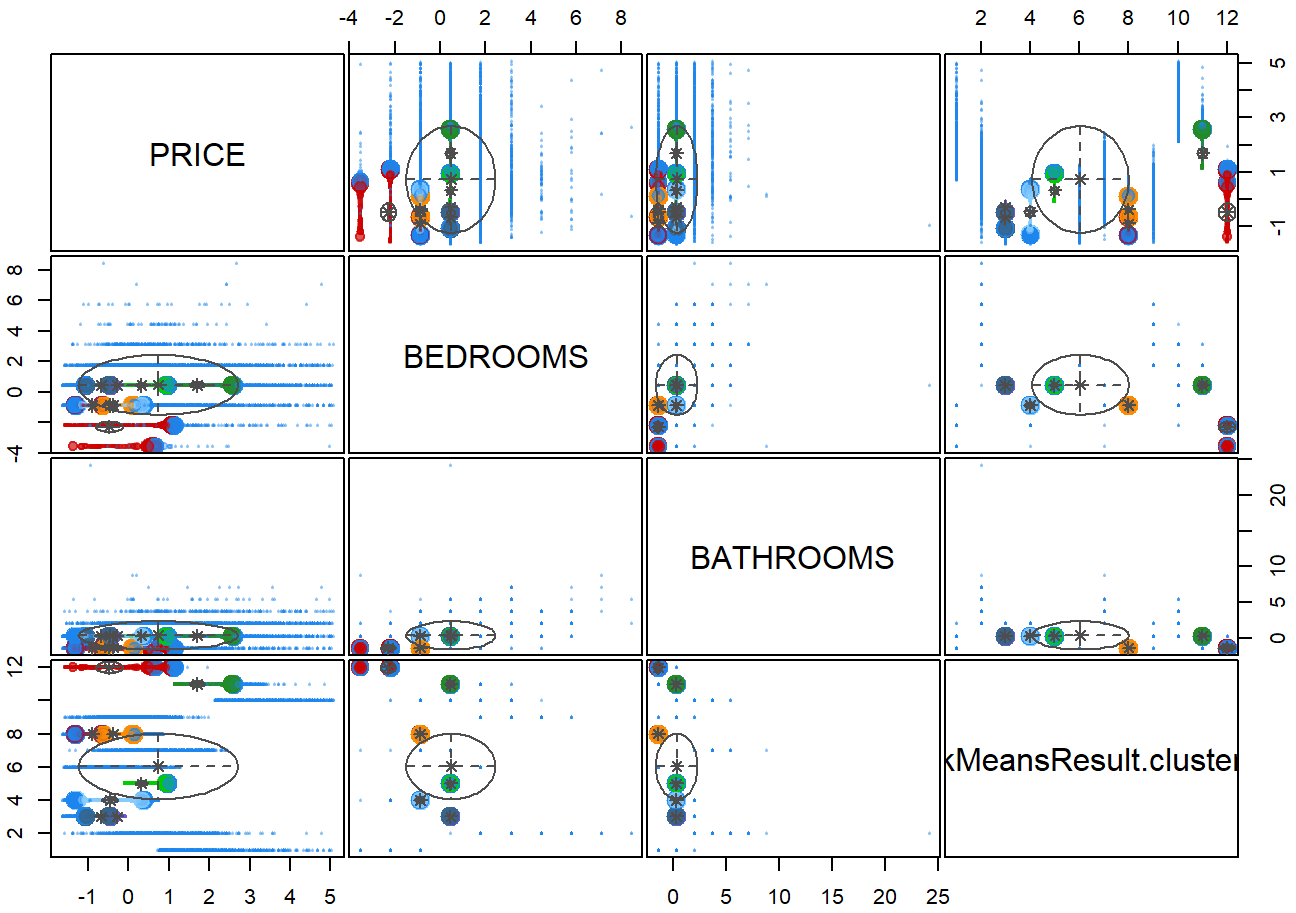
```
summary(modelClust)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VII (spherical, varying volume) model with 9 components:
##
##  log-likelihood     n df       BIC       ICL
##       -64477.04 33656 53 -129506.5 -130976.2
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9
## 9105 1208 4481 4266 2317 4140 2721 1080 4338
```

```
plot(modelClust)
```


BIC
-5e+05
-4e+05
-3e+05
-2e+05
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


PRICE
BEDROOMS
BATHROOMS
kMeansResult.cluster


PRICE
BEDROOMS
BATHROOMS
kMeansResult.cluster


PRICE
BEDROOMS
BATHROOMS
kMeansResult.cluster

Summary

The insights from kMeans Clustering tell us that the clusters are mostly accurate. There may be some other factors as I listed before that can have an effect on the price of a hosue such as the location within Perth and the year that it was built, but basing it solely on how many bathrooms and bedrooms a house has with 12 clusters has a fairly high POV of 85.66%. The insights from Hierarchical Clustering tell us that our p-value is low which means that we can reject the null hypothesis, again providing evidence that there is a correlation between price and how many beds and baths a house has. Finally, the results from Model Clustering tell us the clusters are mostly within their covariances of (with most exceptions being from the resulting cluster on the bottom row and right-most column) each other. This provides some evidence that we are not overfitting or underfitting the data as most clusters can be seen in each section.