Navarro_Assignment_3_1

John Navarro

January 22, 2017

1. Perform latent class analysis for market segmentation

First we will download the German Credit data and separate for only categorical variables

```
#Load the German data
germanCredit<- read.csv("C:/Users/JohntheGreat/Documents/MSCA/DataMining/Week1/german_credit.csv", head
#colnames(germanCredit)
#load poLCA
library(poLCA, quietly = TRUE)
```

```
## Warning: package 'poLCA' was built under R version 3.3.2
```

```
## Warning: package 'scatterplot3d' was built under R version 3.3.2
```

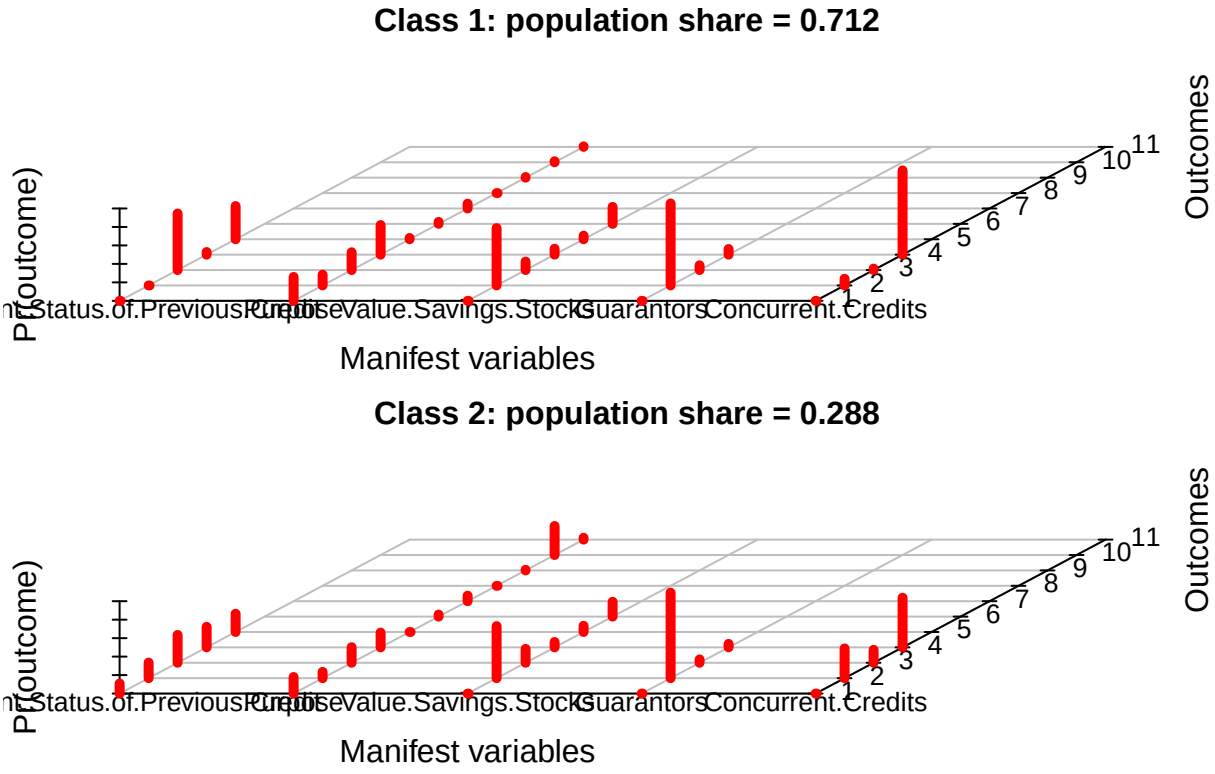
```
# select only the chosen variables
GC.inputs<- germanCredit[, c(4,5,7,11,15)]
colnames(GC.inputs)
```

```
## [1] "Payment.Status.of.Previous.Credit" "Purpose"
## [3] "Value.Savings.Stocks"              "Guarantors"
## [5] "Concurrent.Credits"
```

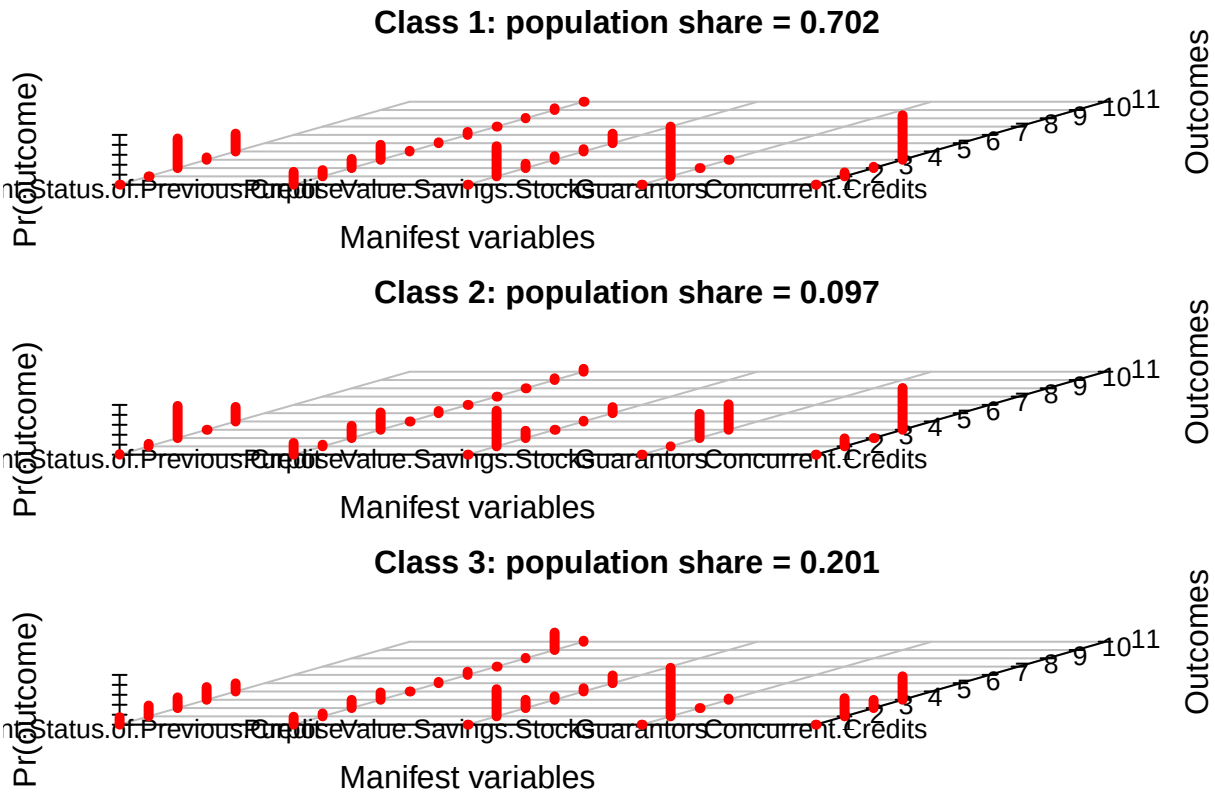
```
#summary(GC.inputs)

# need to split the data into train/test portions
set.seed(555)
train_ind <- sample(seq_len(nrow(GC.inputs)), size = 700)
# separate into two data frames: train and test
train_data <- GC.inputs[train_ind, ]
test_data<- GC.inputs[-train_ind, ]
train_alt <- train_data +1
test_alt <- test_data +1
```

2. Determine K cluster solutions

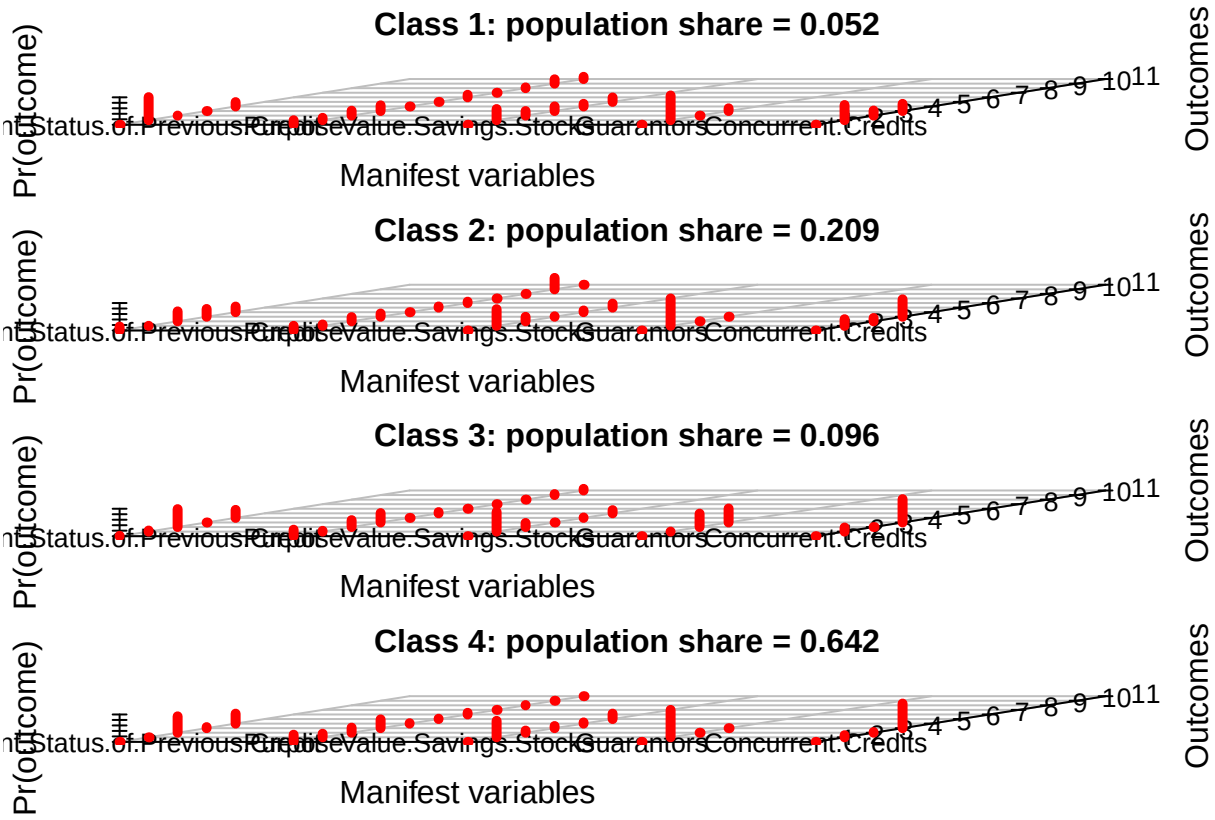
```
#Run LCA
set.seed(555)
f1 = cbind(Payment.Status.of.Previous.Credit, Purpose, Value.Savings.Stocks, Guarantors,Concurrent.Credi
results.2 <- poLCA(f1,train_alt,nclass=2,nrep=500,tol=.001,verbose=FALSE,graphs=TRUE)
```



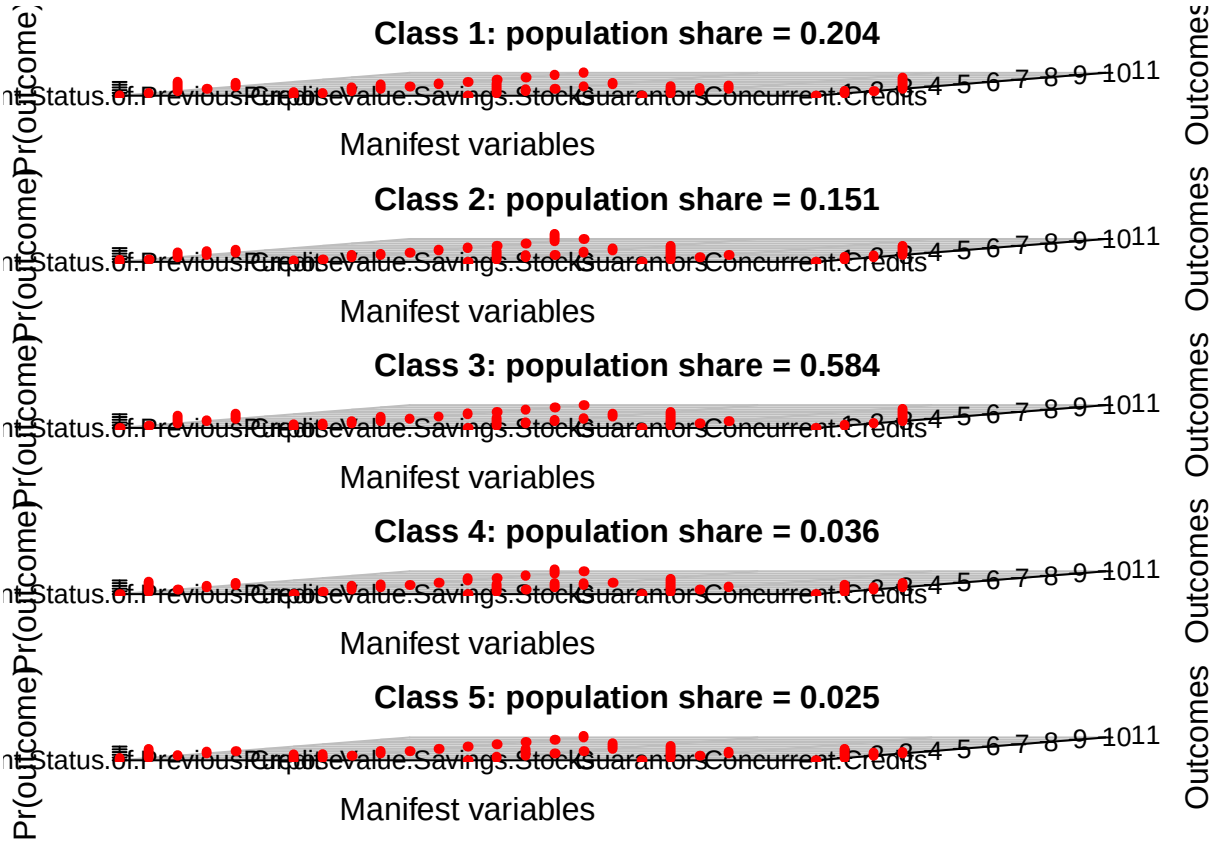

```
results.3 <- poLCA(f1,train_alt,nclass=3,nrep=500,tol=.001,verbose=FALSE,graphs=TRUE)
```




```
results.4 <- poLCA(f1,train_alt,nclass=4,nrep=500,tol=.001,verbose=FALSE,graphs=TRUE)
```




```
results.5 <- poLCA(f1,train_alt,nclass=5,nrep=500,tol=.001,verbose=FALSE,graphs=TRUE)
```




```
results.3$npar
```

```
## [1] 77
```

```
table(results.3$predclass)
```

```
##
##   1   2   3
## 528  68 104
```

```
# Compare the AIC values for each group of clusters
c(results.2$aic,results.3$aic,results.4$aic,results.5$aic)
```

```
## [1] 7248.342 7233.005 7239.064 7256.281
```

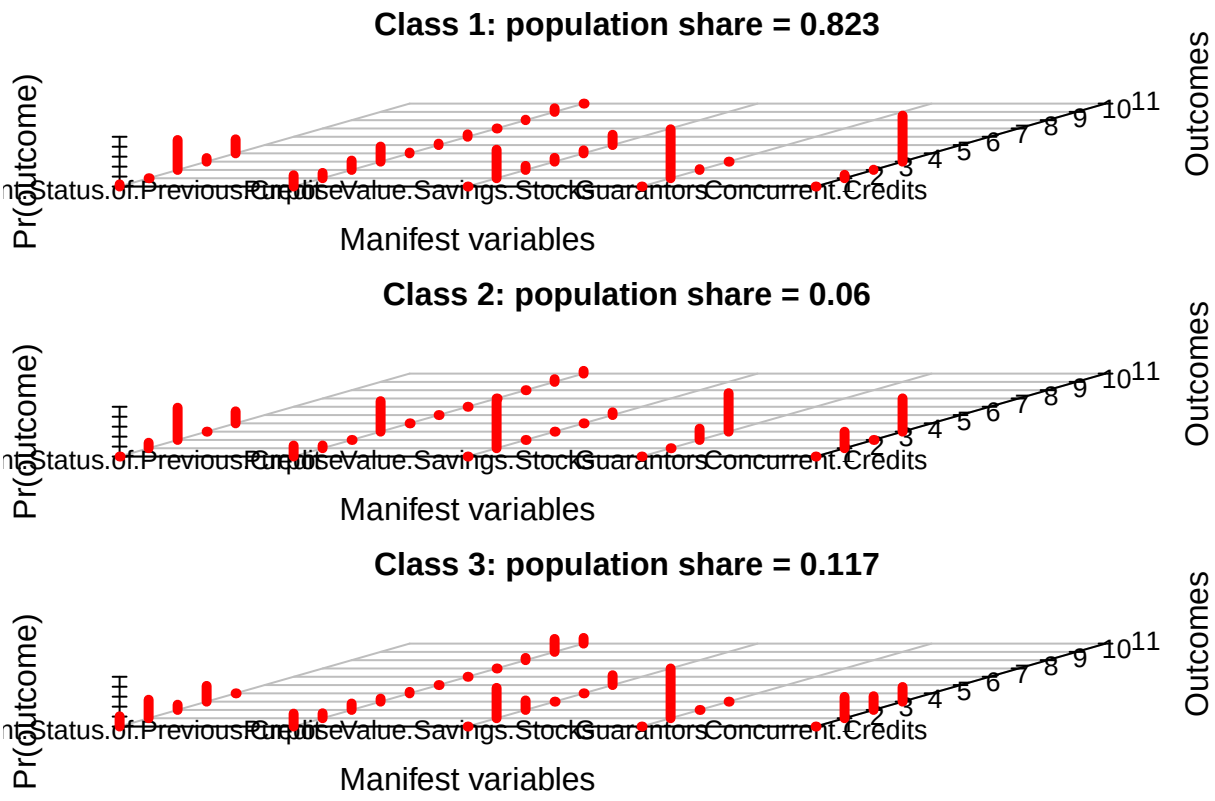
```
c(results.2$bic,results.3$bic,results.4$bic,results.5$bic)
```

```
## [1] 7480.447 7583.439 7707.825 7843.371
```

Here we see that AIC recommends using 3 clusters, since it returns the lowest AIC values for results 2 through 5. Similarly, the graphs show us that separating into 3 clusters is the best choice. The 4 cluster grouping has one group that only contains 5% of the population and 5 clusters has 2 classes with less than 4% of the population share. The 3 cluster grouping has appropriate population share distributions, as well as distinct groupings in regards to all variables, except for Value.Savings.Stocks.

3. Perform Holdout validation of LCA

```
# Perform holdout testing using the centers from training set
f1 = cbind(Payment.Status.of.Previous.Credit, Purpose, Value.Savings.Stocks, Guarantors,Concurrent.Credi
results.3.ho <- poLCA(f1,test_alt,nclass=3,nrep=1,tol=.001,verbose=FALSE,graphs=TRUE,probs.start=results
```




```
table(results.3.ho$predclass)
```

```
##
##   1   2   3
## 250  19  31
```

```
#Look at relative class sizes and conditional probabilites
results.3$probs
```

```
## $Payment.Status.of.Previous.Credit
##                  Pr(1)        Pr(2)     Pr(3)         Pr(4)     Pr(5)
## class 1:  8.308667e-05 1.379751e-18 0.5942707  4.695851e-02 0.3586877
## class 2:  1.063807e-02 5.098722e-02 0.6457623 9.081280e-313 0.2926124
## class 3:  1.507552e-01 2.168248e-01 0.2146831  2.550017e-01 0.1627353
##
## $Purpose
##               Pr(1)      Pr(2)     Pr(3)     Pr(4)         Pr(5)
## class 1:  0.2574267 0.12518394 0.1834525 0.3006452  1.424534e-02
## class 2:  0.2350232 0.03296003 0.2462274 0.3479824 6.789889e-302
## class 3:  0.1603148 0.05858562 0.1645116 0.1469151 4.522026e-106
##                Pr(6)         Pr(7) Pr(8)         Pr(9)     Pr(10)
```

```
## class 1:  0.01711152  5.919176e-02     0  7.739089e-03 0.03500398
## class 2:  0.04388816 3.304923e-296     0 4.449412e-318 0.03490970
## class 3:  0.02568874  6.327167e-02     0  8.497178e-03 0.35089708
##                Pr(11)
## class 1:  4.643048e-98
## class 2:  5.900914e-02
## class 3:  2.131822e-02
##
## $Value.Savings.Stocks
##           Pr(1)     Pr(2)      Pr(3)         Pr(4)      Pr(5)     Pr(6)
## class 1:      0 0.6056586 0.08681097  7.013204e-02 0.04646587 0.1909325
## class 2:      0 0.7164207 0.14700443 3.589254e-297 0.01477926 0.1217956
## class 3:      0 0.5455879 0.16599407  6.770140e-02 0.06506304 0.1556536
##
## $Guarantors
##           Pr(1)        Pr(2)        Pr(3)        Pr(4)
## class 1:      0 1.0000000000 3.724525e-11 2.584892e-25
## class 2:      0 0.0009890859 4.872335e-01 5.117774e-01
## class 3:      0 0.9763088028 2.214231e-28 2.369120e-02
##
## $Concurrent.Credits
##           Pr(1)      Pr(2)         Pr(3)     Pr(4)
## class 1:      0 0.07883942  2.630312e-02 0.8948575
## class 2:      0 0.16025788 2.241696e-159 0.8397421
## class 3:      0 0.36488046  1.637893e-01 0.4713302
```

```
results.3.ho$probs
```

```
## $Payment.Status.of.Previous.Credit
##                   Pr(1)        Pr(2)     Pr(3)         Pr(4)        Pr(5)
## class 1:   4.379641e-02 1.767542e-14 0.6016387  7.259219e-02 2.819727e-01
## class 2:  2.413631e-272 1.104936e-01 0.6481484 1.829927e-270 2.413580e-01
## class 3:   2.055371e-01 3.718952e-01 0.1057748  3.167930e-01 4.699992e-17
##
## $Purpose
##               Pr(1)      Pr(2)        Pr(3)      Pr(4)      Pr(5)
## class 1:  0.2264936 0.10736860 1.882688e-01 0.30272524 0.01601191
## class 2:  0.2209875 0.05524685 5.051059e-10 0.61327199 0.00000000
## class 3:  0.2594375 0.09972630 1.289696e-01 0.06131785 0.02992226
##                   Pr(6)        Pr(7) Pr(8)       Pr(9)     Pr(10)
## class 1:   2.834659e-02 4.859415e-02     0 0.009739546 0.07245156
## class 2:  4.379002e-293 0.000000e+00     0 0.000000000 0.05524683
## class 3:   1.617408e-09 5.437885e-28     0 0.045625370 0.26057180
##                  Pr(11)
## class 1:  7.227667e-132
## class 2:   5.524681e-02
## class 3:   1.144293e-01
##
## $Value.Savings.Stocks
##           Pr(1)     Pr(2)         Pr(3)        Pr(4)         Pr(5)
## class 1:      0 0.5732552  8.223339e-02 7.694074e-02  6.074269e-02
## class 2:      0 0.9447531 7.520149e-260 0.000000e+00 6.864051e-289
## class 3:      0 0.6104191  1.914693e-01 9.323026e-12  1.144684e-22
##                Pr(6)
## class 1:  0.20682799
```

```
## class 2:  0.05524691
## class 3:  0.19811168
##
## $Guarantors
##           Pr(1)        Pr(2)        Pr(3)        Pr(4)
## class 1:      0 9.842092e-01 1.579079e-02 8.331833e-35
## class 2:      0 2.084139e-06 2.265432e-01 7.734547e-01
## class 3:      0 1.000000e+00 4.896661e-44 3.166628e-42
##
## $Concurrent.Credits
##           Pr(1)      Pr(2)       Pr(3)     Pr(4)
## class 1:      0 0.06884333 0.005445893 0.9257108
## class 2:      0 0.33148147 0.000000000 0.6685185
## class 3:      0 0.42909795 0.276208657 0.2946934
```

4. Provide implications on the solutions

The results from the test data seem to match the class characteristics from the training data set. They have similar population proportions train(70%, 20% 10%) and test(82%, 12%, 6%). Furthermore, the probability distributions for each variable per class are similar between both datasets.

5. Comment on the similarities and differences between the clustering solutions from Assignment 2 with the solution you generated using LCA

K-means and KO-means were the two clustering techniques from last week. They were similar to LCA in the sense that they both were able to cluster the data set into different classes. The techniques allowed you to study them and choose a optimal number of classes. For K-means and KO-means we used scree plots to look at the point where additional classes do not add much in the way of a higher VAF. Similarly, we were able to choose an optimal number of classes in LCA by plotting AIC. Here we look for the minimal AIC value to give us an indication of how many clusters to proceed with.

Despite these small similarities, the two techniques are very distinct. First, I chose different variables to represent the data. With K-means and KO-means I chose 4 numeric variables that were scaled to represent the data. While using LCA, I used 5 different variables that were all categorical. This allowed K-means and KO-means to be easier to interpret. We could describe the classes based on their attributes in the chosen variables. LCA is more difficult to interpret, since it is reliant on the existence of an unobserved latent variable that is affecting the observed variables.

Since we randomly split the data in both clustering techniques, we can say that both models were stable since they returned similar clusters in both the training and test data sets.