Data Mining - Assignment 1

Ryan Durfey

Thursday, January 22, 2015

In performing cluster analysis on the GermanCredit data for market segmentation, we first will limit the variables considered to "Age" and "Amount". The variable selection process first limited the field to only numeric variables, for which the K-Means method is well-suited. Secondly, the ages of and loan amounts of the observations (people) are of personal interest, so they were chosen for this analysis.

```
require(caret)
require(dplyr)
require(knitr)
require(NbClust)
source("./komeans.R")
```

```
set.seed(132435)

data(GermanCredit)
GC<-GermanCredit

## selecting numeric variables of interest, Age and Amount
GC<-GC[,c("Amount","Age")]

## separate training and test datasets to encompass 70% and 30% of the data, respectively
GC_train<-sample_frac(GC,.70)
GC_test<-setdiff(GC,GC_train)
dim(GC_train)
```

```
## [1] 700   2
```

```
dim(GC_test)
```

```
## [1] 300   2
```

```
## making sure none of the rows intersect, confirming the dataset separation was successful
nrow(intersect(GC_train,GC_test))
```

```
## [1] 0
```

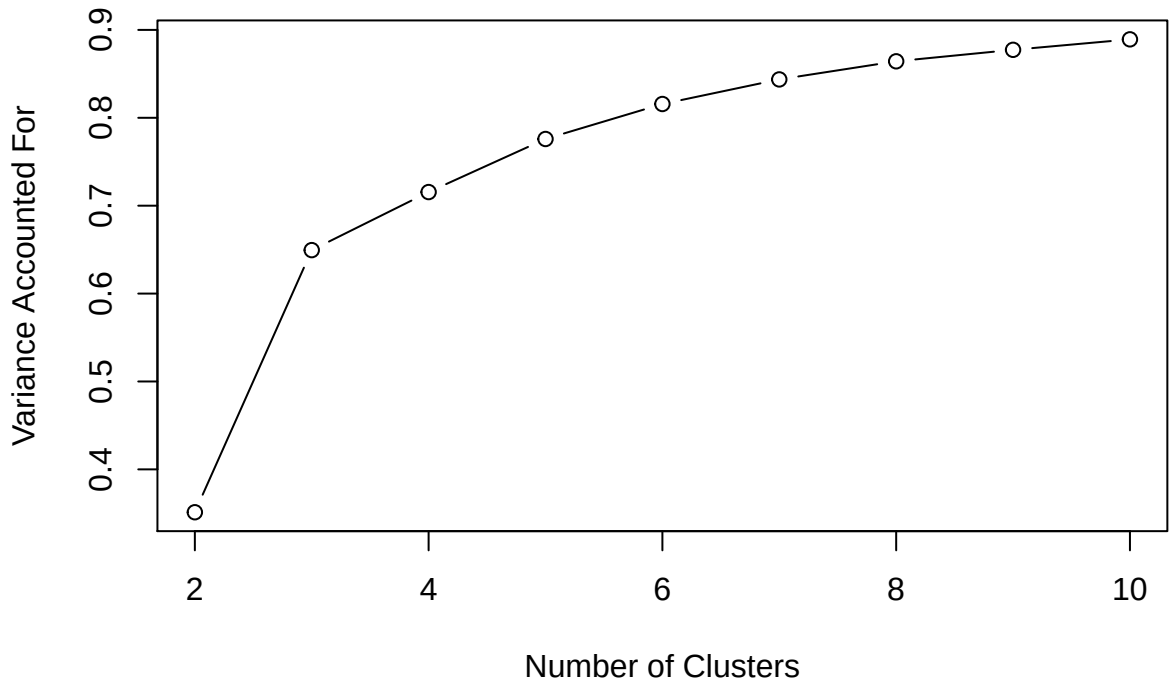
```
## scale the data
GC_train_unscaled<-GC_train
GC_test_unscaled<-GC_test
GC_train<-scale(GC_train)
GC_test<-scale(GC_test)

## run kmeans on training data in iterative function form
VAFs<-function(data,nc.min=2,nc.max=10,seed=132435){
        set.seed(seed)
        vaf<-0
        for(i in nc.min:nc.max){
```

```
                k<-kmeans(data,centers=i,nstart=100)
                vaf[i]<-k$betweenss/k$totss
        }
        vaf[-1]
}
VAF<-VAFs(GC_train)
## variance accounted for
kable(cbind(num.clusters=2:10,variance.accounted.for=VAF),caption="Variance Accounted for
      by Number of Clusters Used")
```

num.clusters	variance.accounted.for
2	0.3512169
3	0.6494306
4	0.7154905
5	0.7758721
6	0.8156781
7	0.8436606
8	0.8642600
9	0.8773288
10	0.8892398

Table 1: Variance Accounted for by Number of Clusters Used

```
## Scree Plot
plot(2:10,VAF,type="b",xlab="Number of Clusters",ylab="Variance Accounted For")
```




```
## There is an obvious '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 and K-Overlapping-Means analyses.

```
## After choosing 3 Clusters, running kmeans again
K3<-kmeans(GC_train,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")
```

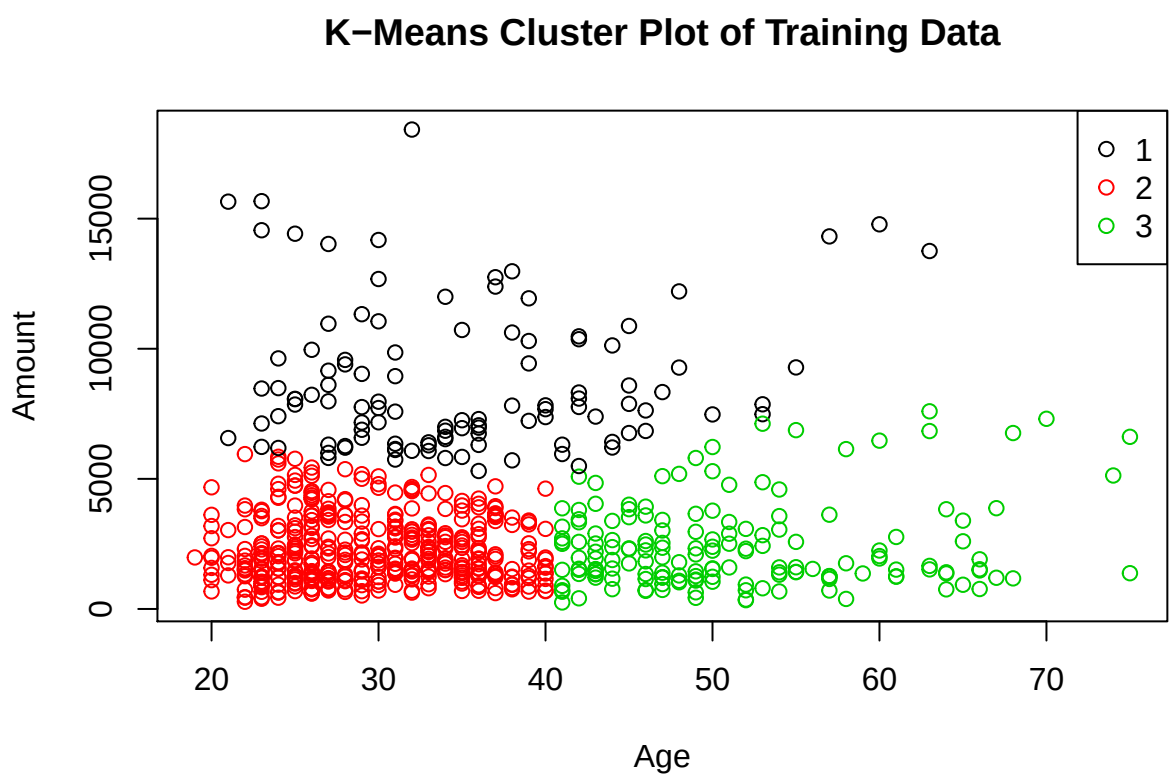
Cluster	# of Obs.
1	110
2	426
3	164

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

```
## Variance Accounted For
K3_VAF<-K3$betweenss/K3$totss
K3_VAF
```

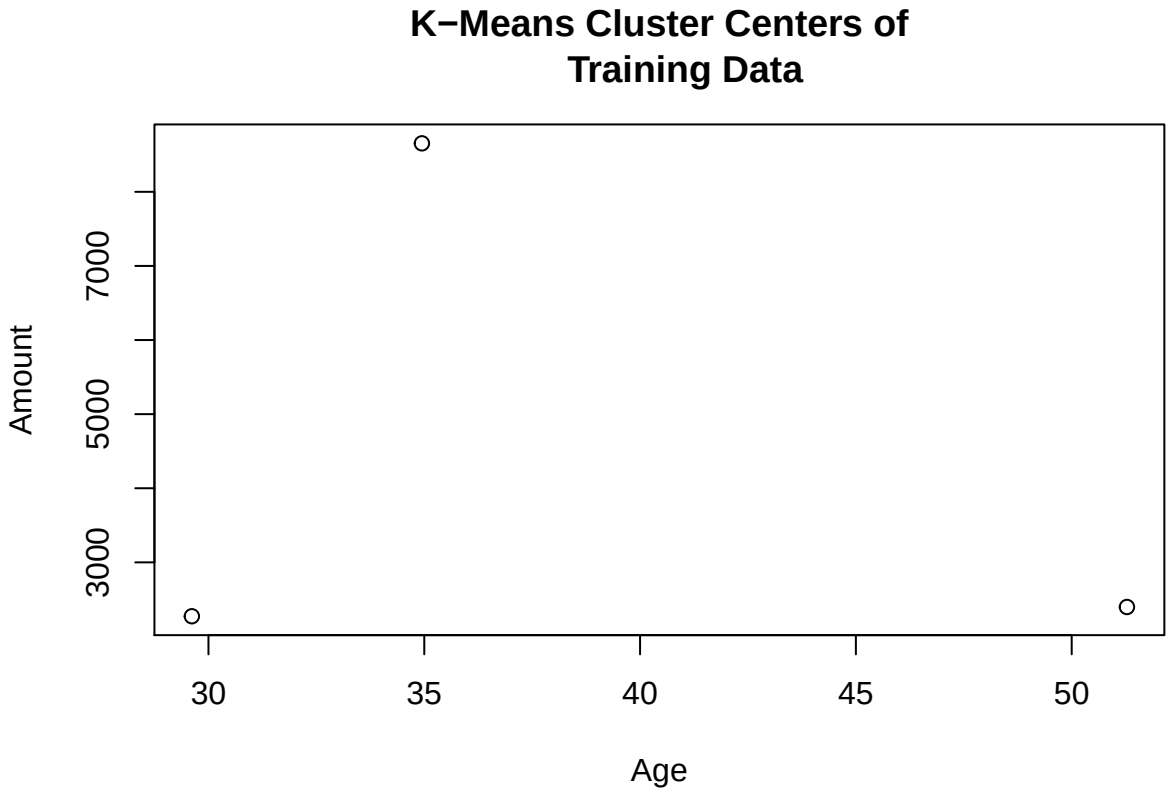
```
## [1] 0.6494306
```

```
## cluster plots of training data and centers of clusters. unscaled data
plot(GC_train_unscaled[,2],GC_train_unscaled[,1],col=K3$cluster,xlab="Age",ylab="Amount",
     main="K-Means Cluster Plot of Training Data")
legend("topright",legend=as.data.frame(table(K3$cluster))[,1],
       col=as.data.frame(table(K3$cluster))[,1],cex=1,pch=1)
```



In the above plot, we see each cluster representing different regions. In the bottom-left cluster (red) are observations that are comprised of fairly young people (below ~42) with fairly small loan amounts (less than ~5000). The bottom-right cluster (green) appears similar to the red cluster except for older people (above ~42). The top cluster (black) looks to incorporate the higher loan amounts (more than ~7000) with its mean age likely falling somewhere in the 30's.

```
unscld_age<-tapply(GC_train_unscaled[,"Age"],K3$cluster,mean)
unscld_amt<-tapply(GC_train_unscaled[,"Amount"],K3$cluster,mean)
plot(unscld_age,unscld_amt,xlab="Age",ylab="Amount",main="K-Means Cluster Centers of
     Training Data")
```



This second plot, showing the centers of the K-Means model of the training data set, shows us what the previous one hinted at. A center describing young people and small loan amounts, a center showing older people with small to medium loan amounts, and a center with large loan amounts for peolpe in their mid-30s.

Now that we have our model created using K-Means on the training dataset, we can perform the holdout by applying the model to the test dataset by using the same centers.

```
## Holdout
K3_test<-kmeans(GC_test,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")
```

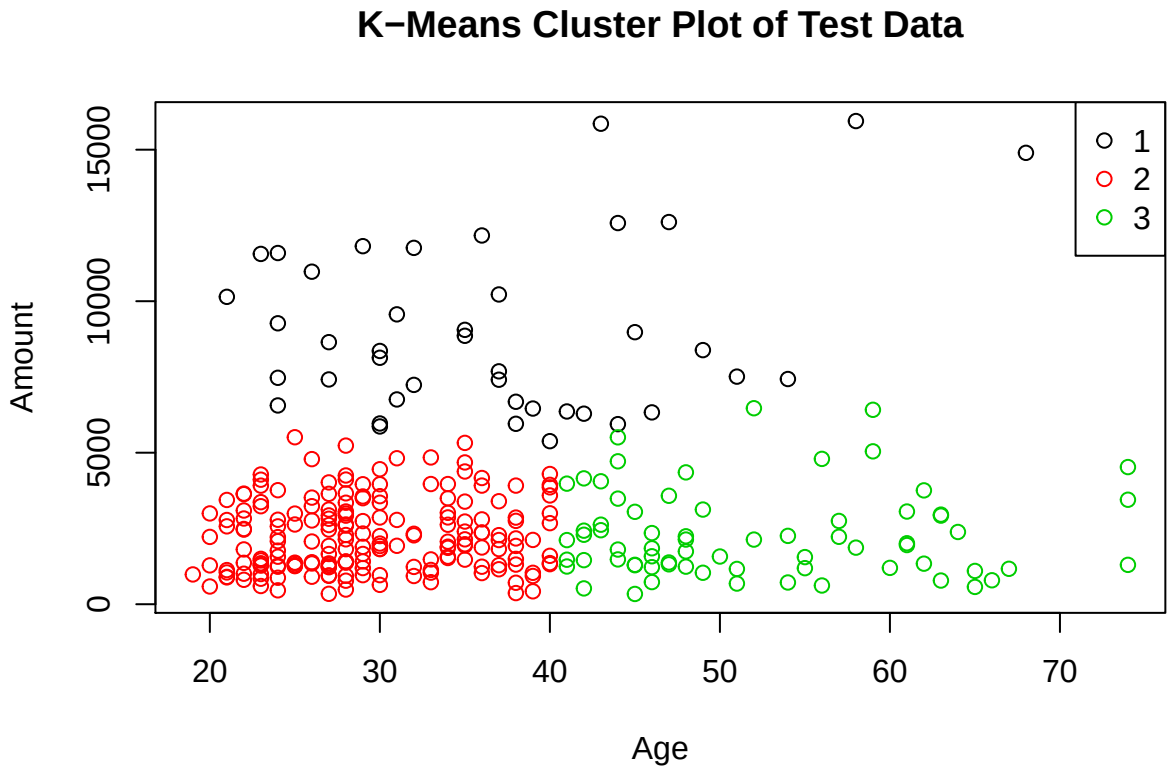
Cluster	# of Obs.
1	41
2	190
3	69

Table 3: No. of Observations per Cluster from Test dataset

```
## Variance Accounted For
K3_test_VAF<-K3_test$betweenss/K3_test$totss
K3_test_VAF
```

```
## [1] 0.654856
```

```
## cluster plot of test data from model. unscaled data
plot(GC_test_unscaled[,2],GC_test_unscaled[,1],col=K3_test$cluster,ylab="Amount",
     xlab="Age",main="K-Means Cluster Plot of Test Data")
legend("topright",legend=as.data.frame(table(K3_test$cluster))[,1],
       col=as.data.frame(table(K3_test$cluster))[,1],cex=1,pch=1)
```




```
## comparison of relative percentage of observations per cluster
kable(cbind(c("Train","Test"),rbind(table(K3$cluster)/length(GC_train[,1]),
      table(K3_test$cluster)/length(GC_test[,1]))),caption="Relative Percentage of
      Observations per Cluster")
```

	1	2	3
Train	0.157142857142857	0.608571428571429	0.234285714285714
Test	0.136666666666667	0.633333333333333	0.23

Table 4: Relative Percentage of Observations per Cluster

We see that our clusters obtained in the Test dataset are not far off from those in the Training dataset, so we will consider the Holdout a success.

Next, we create clusters/partitions using K-Overlapping-Means. We'll use the same amount of clusters as in K-Means, to be able to compare the two. Also for the sake of comparison, the clusters will be made with the Training dataset.

```
## Generating 3 komeans clusters
Ko3<-komeans(data=GC_train,nclust=3,lnorm=2,nloops=100,tolerance=0.001,seed=132435)
## num of obs in each cluster
kable(as.data.frame(table(Ko3$Group+1)),col.names=c("Cluster","# of Obs."),caption="No.
      of Observations per KoMeans Cluster")
```

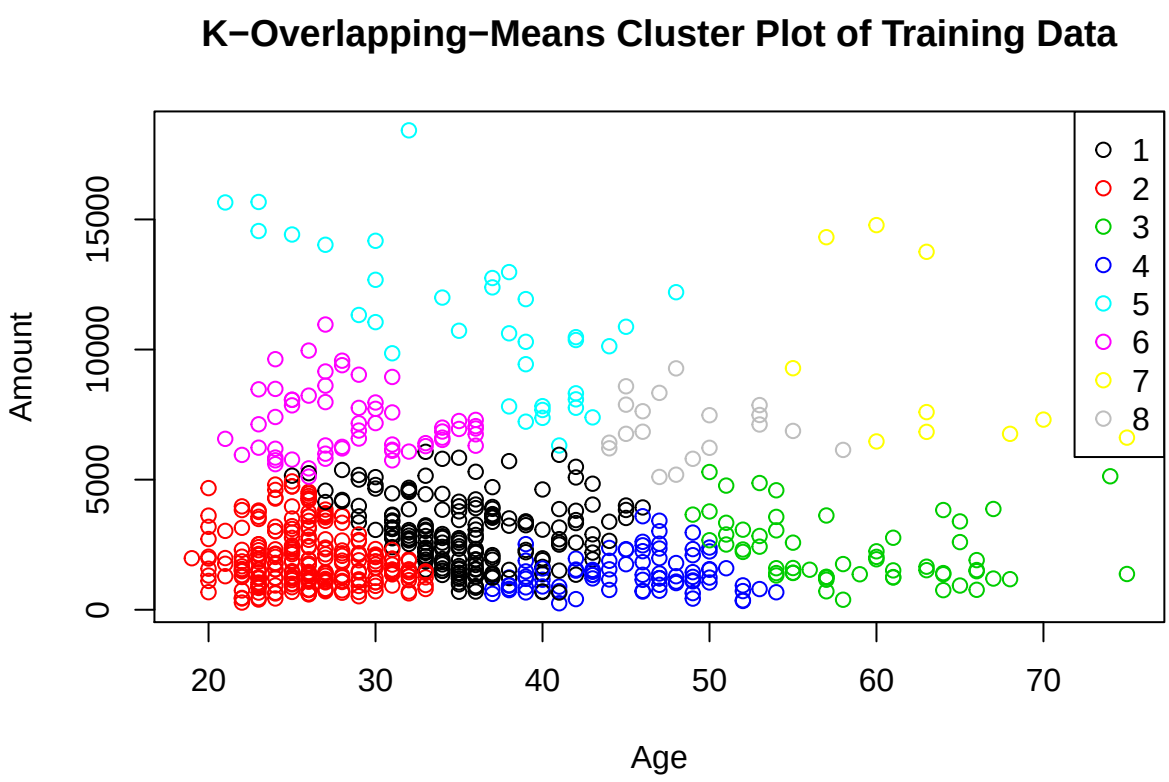
Cluster	# of Obs.
1	190
2	243
3	59
4	85
5	35
6	59
7	10
8	19

Table 5: No. of Observations per KoMeans Cluster

```
## Variance Accounted For
Ko3$VAF
```

```
## [1] 0.7974882
```

```
## plot of KoMeans clusters
plot(GC_train_unscaled[,2],GC_train_unscaled[,1],col=Ko3$Group+1,xlab="Age",ylab="Amount",
     main="K-Overlapping-Means Cluster Plot of Training Data")
legend("topright",legend=as.data.frame(table(Ko3$Group+1))[,1],
       col=as.data.frame(table(Ko3$Group+1))[,1],cex=1,pch=1)
```




```
## comparing clusters from kmeans and komeans
Ka<-as.data.frame(cbind(GC_train,cluster=K3$cluster))
Kb<-as.data.frame(cbind(Ko3$data,group=Ko3$Group+1))
kable(rbind(cbind(Cluster=c(1,2,3),as.data.frame.matrix(table(Ka[,3],Kb[,3]))),
                  Sum=as.data.frame(table(Ka[,3]))[,2]),
            c("Sum",as.data.frame(table(Kb[,3]))[,2],700)),
      caption="Observations Cross-Referenced in Each K-Means and K-Overlapping-Means Cluster")
```

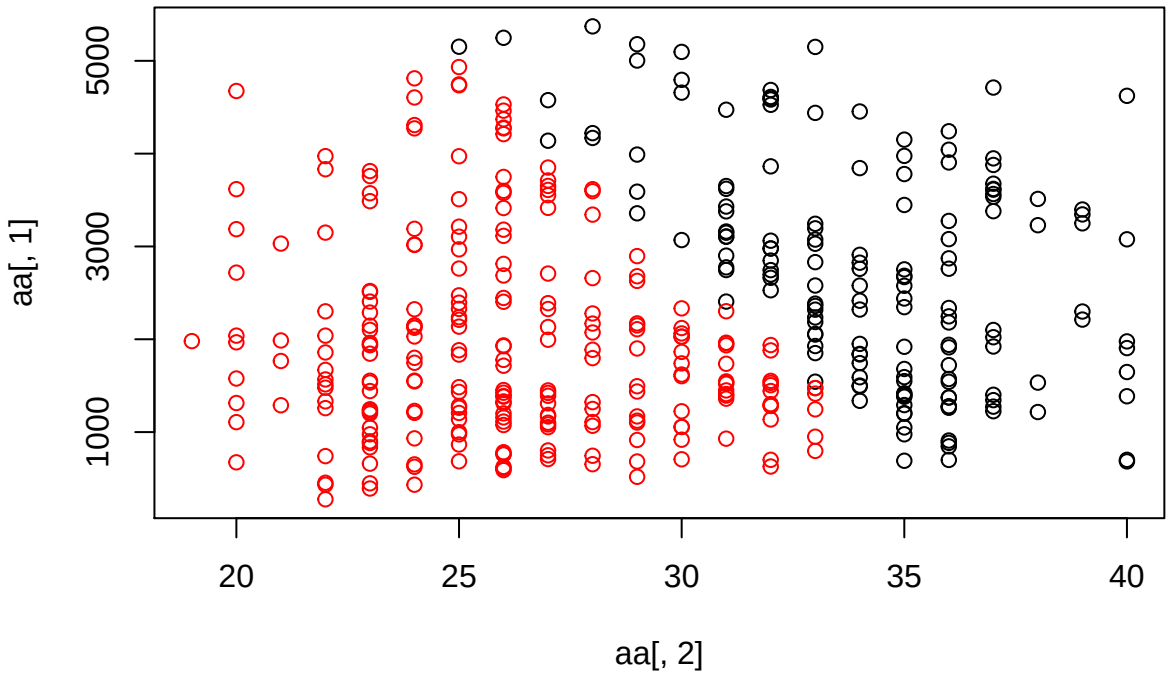
Cluster	1	2	3	4	5	6	7	8	Sum
1	7	0	0	0	35	52	4	12	110
2	158	243	0	18	0	7	0	0	426
3	25	0	59	67	0	0	6	7	164
Sum	190	243	59	85	35	59	10	19	700

Table 6: Observations Cross-Referenced in Each K-Means and K-Overlapping-Means Cluster

From the table above, we see that the cross-referenced regions comprising of K-Means cluster 2 & K-Overlapping-Means partition 1 as well as K-Means cluster 2 & K-Overlapping-Means partition 2 incorporate the highest number of observations: 158 and 243, respectively. The next highest region has 67, so there are significantly more observations within these cluster regions than in any other.

To further analyze these two regions, we can determine the ranges of their Age and Amount values.

```
a<-cbind(GC_train_unscaled,cluster=K3$cluster,group=Ko3$Group+1)
a1<-a[a$cluster==2 & a[,4]==1,]
a2<-a[a$cluster==2 & a[,4]==2,]
aa<-rbind(a1,a2)

## plot of two regions
plot(aa[,2],aa[,1],col=aa[,4])
```




```
## table of partition encompassing KMeans cluster 2 and KoMeans partition 1
kable(rbind(Age=range(a1[,2]),Amount=range(a1[,1])),col.names=c("Min","Max"),
      caption="Ranges from Cluster 2 & Partition 1")
```

	Min	Max
Age	25	40
Amount	684	5371

Table 7: Ranges from Cluster 2 & Partition 1

```
## table of partition encompassing KMeans cluster 2 and KoMeans partition 2
kable(rbind(Age=range(a2[,2]),Amount=range(a2[,1])),col.names=c("Min","Max"),
      caption="Ranges from Cluster 2 & Partition 2")
```

	Min	Max
Age	19	33
Amount	276	4933

Table 8: Ranges from Cluster 2 & Partition 2

From our market segmentation, these two regions of observations, with respect to their Age and Loan Amount, are of the highest interest to us. For the purpose of creating focus groups for use in Attitudinal and Usage studies, we would like to focus on people that fall within these segments because they comprise the largest number of loans. Thus, understanding these segments in particular is more valuable than other segments.

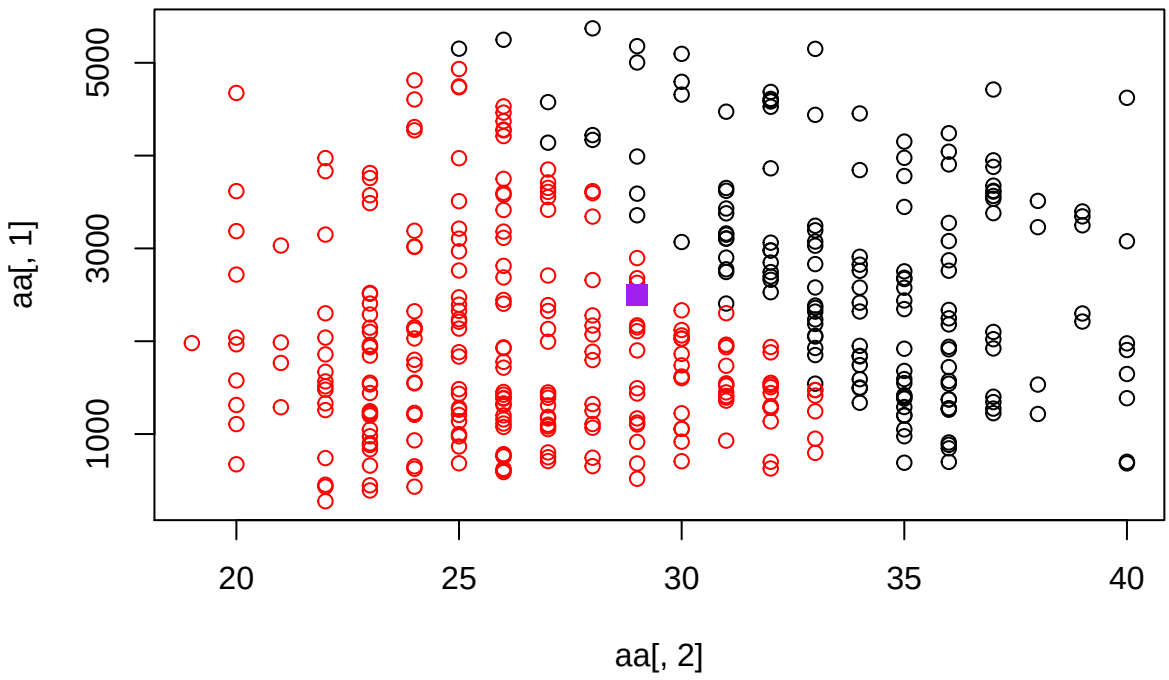
For recruiting people into such a A&U study focus groups, we can identify potential candidates first by choosing people that fall within those ranges. For example, a person 25-41 years old and who has a loan amount between 609 and 4,844 is a candidate for the first segment.

It should be noted that the ranges of both segments overlap significantly. Thus, to determine to which segment they will belong, we can choose the shortest euclidean distance to each segments' center. An example of this is below.

```
Candidate_Age<-29
Candidate_Amount<-2500
Candidate<-c(Age=Candidate_Age,Amount=Candidate_Amount)
seg1_center<-c(Age=mean(a1[,2]),Amount=mean(a1[,1]))
seg2_center<-c(Age=mean(a2[,2]),Amount=mean(a2[,1]))

kable(cbind(dist(rbind(Candidate,seg1_center)),dist(rbind(Candidate,seg2_center))),
      col.names=c("Distance to Segment 1","Distance to Segment 2"))
```

Distance to Segment 1	Distance to Segment 2
240.8033	546.0982

```
## plot of two regions with example point in purple
plot(aa[,2],aa[,1],col=aa[,4])
points(Candidate_Age,Candidate_Amount,type="p",col="purple",pch=15,cex=1.5)
```



In the above example, such a candidate would fall into the first segment (representing the area comprised of cluster 2 and partition 1 from the previous K-Means and K-Overlapping-Means analyses).

As for implementing these focus groups, we would first identify potential candidates from our database that fit the above criteria. Then we could randomly sample 30 per each segment and call them to request their participation in the study. For any amount 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.