

Class Project #2:
Exploring Variations in Clustering and Predictive Analysis

Functions used in the project:

Functions:
read.csv()
head()
subset()
na.omit()
iclust()
kmeans()
lm()
glm()
summary()
pairs()
predict()
head()
as.data.frame()
is.na()
sample()
select()
scale()
summarise()
as.matrix()
gsub()

factor()
as.numeric()
kGmedian()
plot()
fit()

In the project, we have taken subsets of Occupation 0, Marital Status 1 with age 45 and City Category A with age 25 for clustering methods and linear modeling.

To make the DataSet numeric

For making the data set numeric, we used the below code. For productid, p was converted to 1. City Category A, B and C were numbered 1,2 and 3 respectively. Marital Status was 1 and 0 for married and unmarried. Gender was 1 and 0 for female and male. For age, the highest number from the range was taken.

```
Data <- read.csv(file="BlackFriday.csv", header=TRUE, sep=",")
Data.clean <- na.omit(Data)
```

```
#to change City types "A" to numeric values "1"
Data.clean$City_Category<-factor(Data.clean$City_Category)
Data.clean$City_Category<-as.numeric(Data.clean$City_Category)
```

```
#to change Gender types "M" to numeric value "1"
Data.clean$Gender<-factor(Data.clean$Gender)
Data.clean$Gender<-as.numeric(Data.clean$Gender)
```

```
Data.clean$Age<-gsub("0-17", "17",Data.clean$Age,ignore.case = TRUE)
Data.clean$Age<-gsub("18-25", "25",Data.clean$Age,ignore.case = TRUE)
Data.clean$Age<-gsub("26-35", "35",Data.clean$Age,ignore.case = TRUE)
```

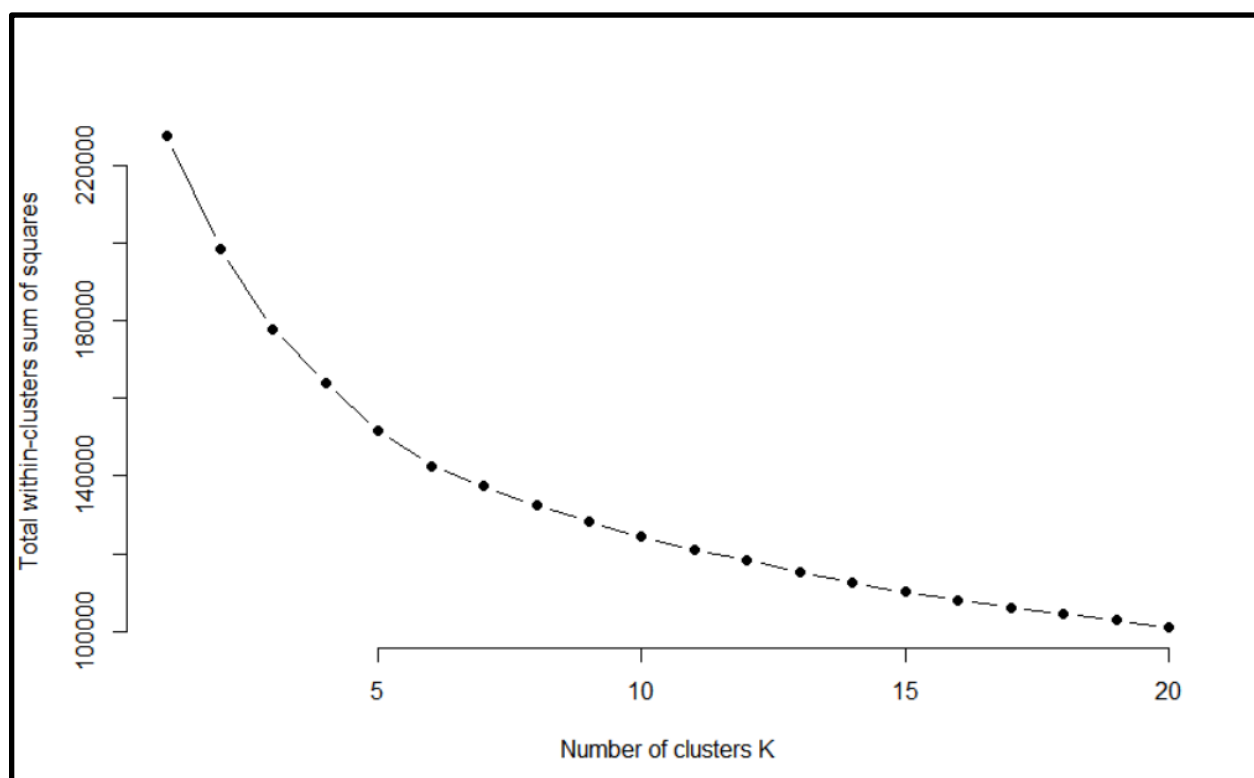
```

Data.clean$Age<-gsub("36-45", "45",Data.clean$Age,ignore.case = TRUE)
Data.clean$Age<-gsub("46-50", "50",Data.clean$Age,ignore.case = TRUE)
Data.clean$Age<-gsub("51-55", "55",Data.clean$Age,ignore.case = TRUE)
Data.clean$Age<-gsub("55[+]", "56",Data.clean$Age,ignore.case = TRUE)
Data.clean$Product_ID<-gsub("P", "1",Data.clean$Product_ID,ignore.case = TRUE)
Data.clean$Stay_In_Current_City_Years<-gsub("[+]",
"",Data.clean$Stay_In_Current_City_Years,ignore.case = TRUE)
Data.clean$Product_ID<-as.numeric(Data.clean$Product_ID)
Data.clean$Age<-as.numeric(Data.clean$Age)
Data.clean$Stay_In_Current_City_Years<-as.numeric(Data.clean$Stay_In_Current_City_Years)

```

The optimal numbers of K:

No of optimal k in Occupation 0:



```

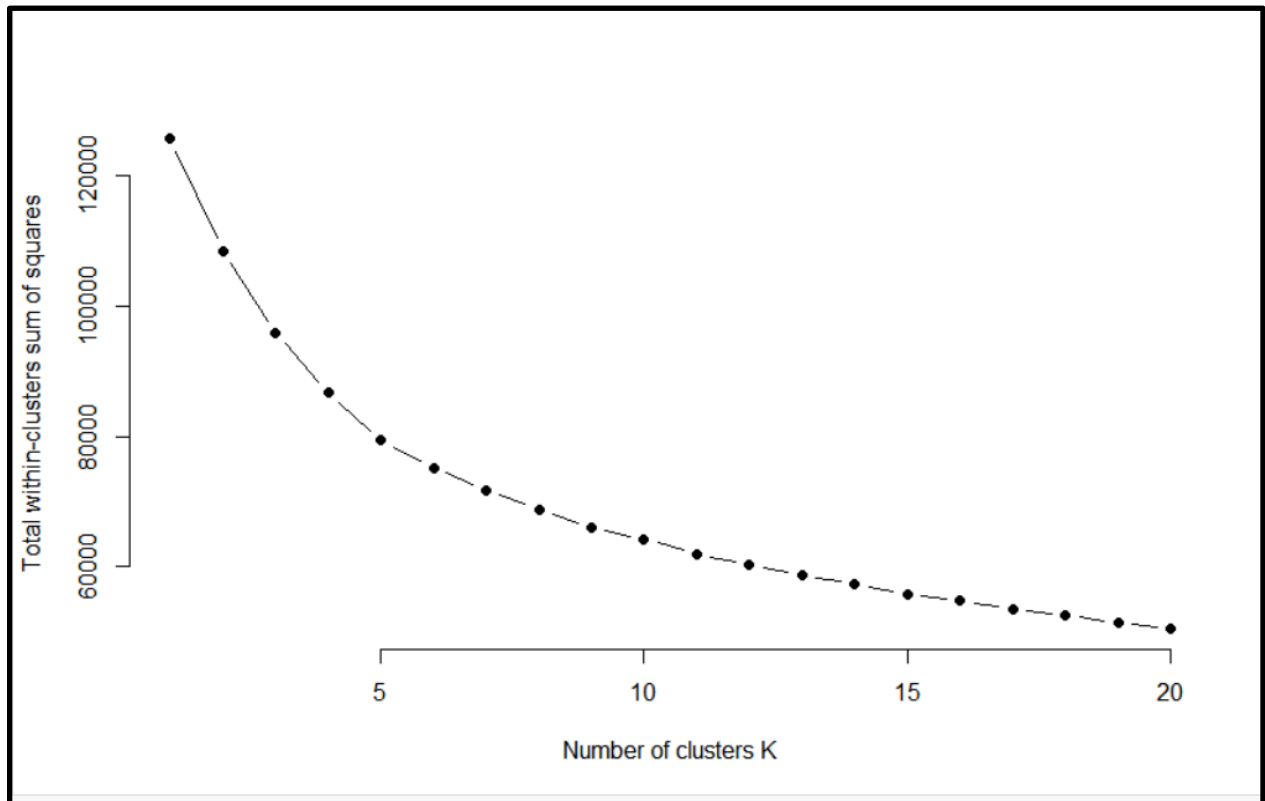
wss <- sapply(1:k.max, function(k) {kmeans(scaled_data.Occ0, k, nstart=25,iter.max = 15)
}$tot.withinss})

```

```
plot(1:k.max, wss, type="b", pch = 19, frame = FALSE, xlab="Number of clusters K",
ylab="Total within-clusters sum of squares")
```

k=4 is the optimal for the Occupation 0.

No of optimal k in Martial Status 1 with Age 45:

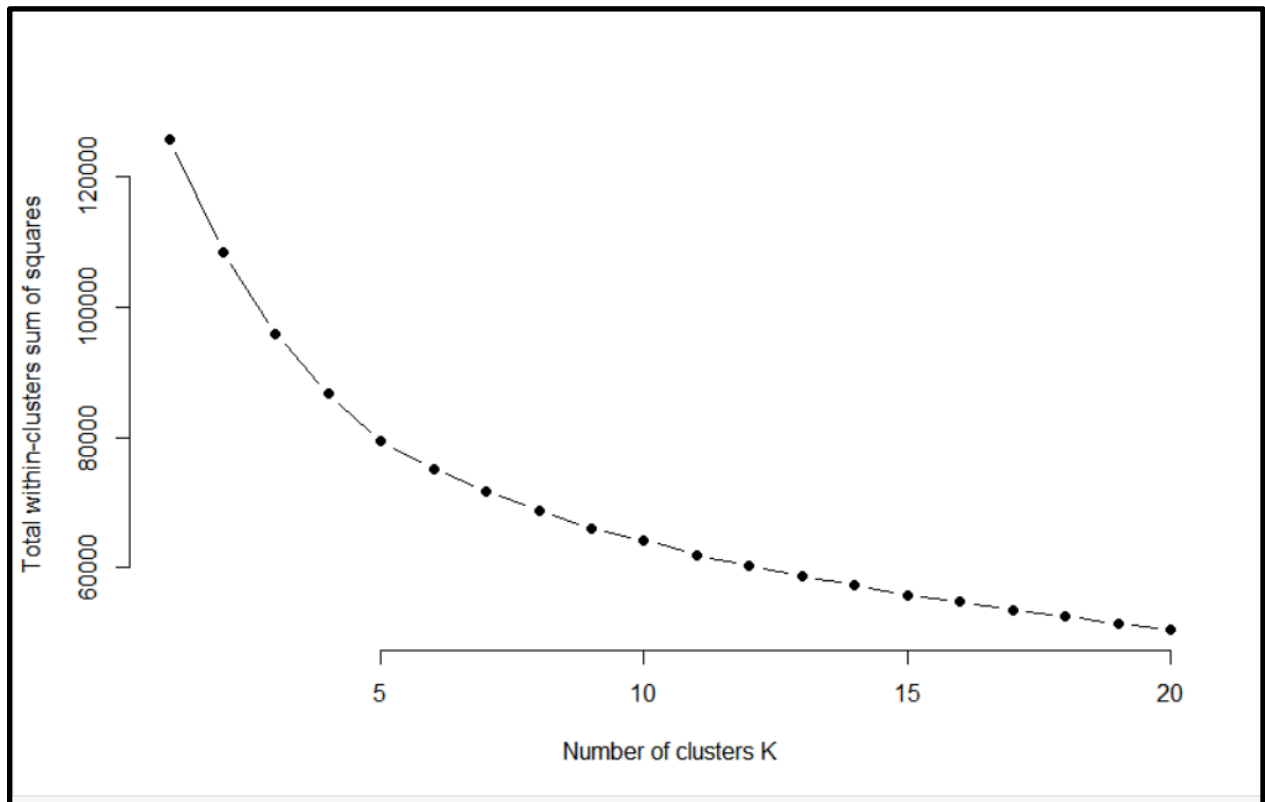


```
wss <- sapply(1:k.max, function(k) {kmeans(scaled_data.Data1.Age45, k, nstart=25, iter.max =
15 )$tot.withinss})
```

```
plot(1:k.max, wss, type="b", pch = 19, frame = FALSE, xlab="Number of clusters K",
ylab="Total within-clusters sum of squares")
```

k=4 is the optimal k for the Marital status 1 with age 45.

No of optimal k in City Category A with Age 25:



```
wss <- sapply(1:k.max, function(k) {kmeans(scaled_data.Data.A.Age25, k, nstart=25, iter.max = 15)$tot.withinss})
```

```
plot(1:k.max, wss, type="b", pch = 19, frame = FALSE, xlab="Number of clusters K",  
ylab="Total within-clusters sum of squares")
```

k=4 is the optimal k for city category A with age 25.

Subsets of Occupation:

#For Clustering#

```
Data <- read.csv(file="BlackFriday1.csv", header=TRUE, sep=",")
```

```
#for omitting NA values
```

```
Data.clean <- na.omit(Data)
```

```
#subsetting for occupation
```

```
Data.Occ0 <- subset(Data.clean, Occupation == "0")
```

```
head(Data.Occ0)
```

```
Data.Occ0$Occupation <- NULL
```

```
Data.Occ.1 <- subset(Data.clean, Occupation == "1")
```

```
Data.Occ2 <- subset(Data.clean, Occupation == "2")
```

```
Data.Occ3 <- subset(Data.clean, Occupation == "3")
```

```
Data.Occ4 <- subset(Data.clean, Occupation == "4")
```

```
Data.Occ5 <- subset(Data.clean, Occupation == "5")
```

```
Data.Occ6 <- subset(Data.clean, Occupation == "6")
```

```
Data.Occ7 <- subset(Data.clean, Occupation == "7")
```

```
Data.Occ8 <- subset(Data.clean, Occupation == "8")
```

```
Data.Occ9 <- subset(Data.clean, Occupation == "9")
```

```
Data.Occ10 <- subset(Data.clean, Occupation == "10")
```

```
Data.Occ11 <- subset(Data.clean, Occupation == "11")
```

```
Data.Occ13 <- subset(Data.clean, Occupation == "13")
```

```
Data.Occ14 <- subset(Data.clean, Occupation == "14")
```

```
Data.Occ15 <- subset(Data.clean, Occupation == "15")
```

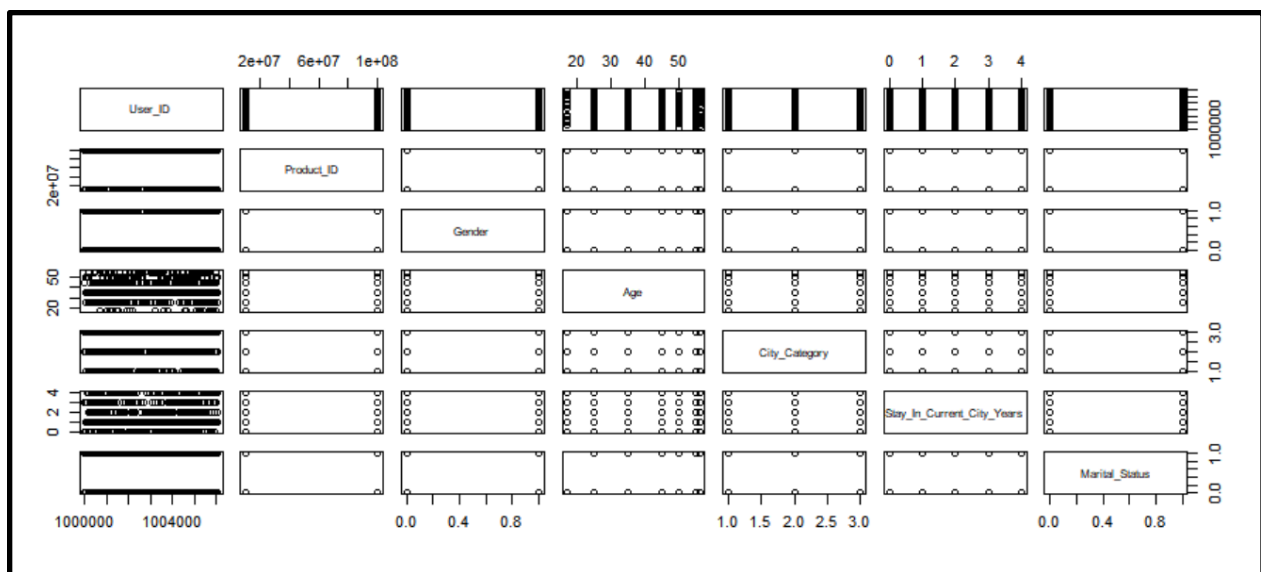
```
Data.Occ16 <- subset(Data.clean, Occupation == "16")
```

```
Data.Occ17 <- subset(Data.clean, Occupation == "17")
```

```
Data.Occ18 <- subset(Data.clean, Occupation == "18")
```

```
Data.Occ19 <- subset(Data.clean, Occupation == "19")
```

```
Data.Occ20 <- subset(Data.clean, Occupation == "20")
```

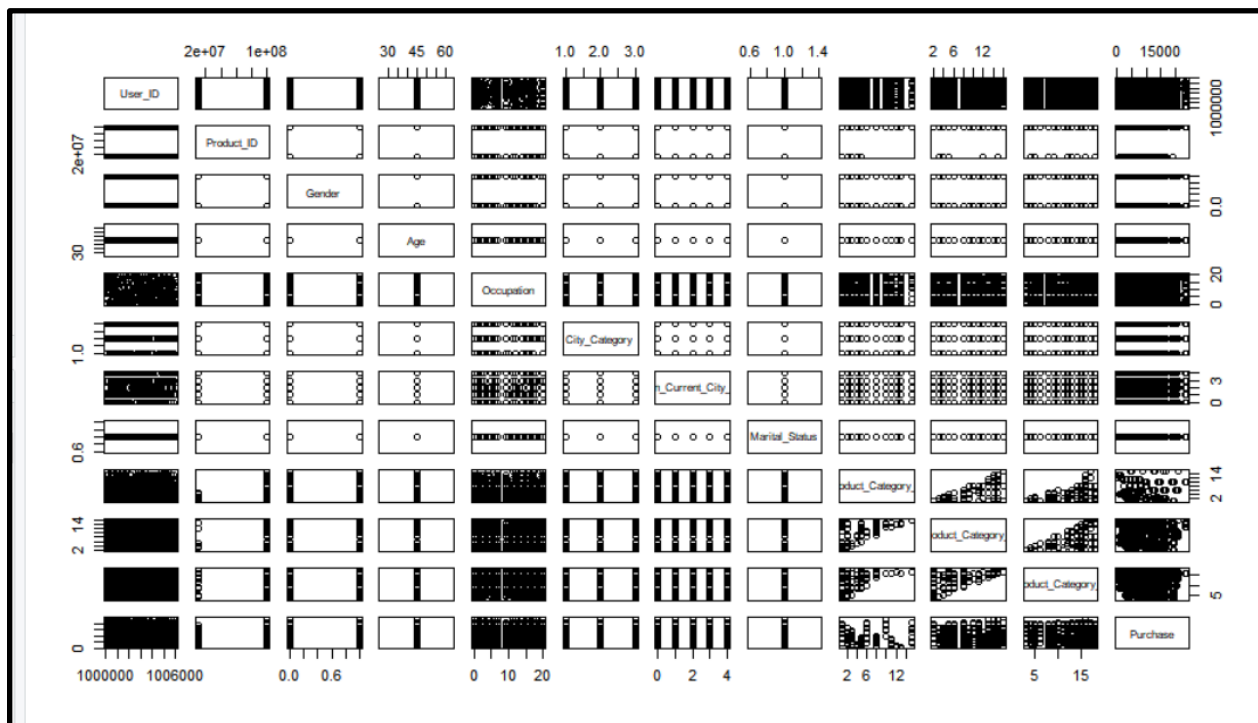
Pairwise Plotting for Occupation 0:

Subsets of Marital Status and then by Age:

#subset of marital status and then subsetting by age

```
Data.Mart1 <- subset(Data.clean, Marital_Status == "1")
Data1.Age17 <- subset(Data.Mart1, Age=="17")
Data1.Age25 <- subset(Data.Mart1, Age=="25")
Data1.Age35 <- subset(Data.Mart1, Age=="35")
Data1.Age45 <- subset(Data.Mart1, Age=="45")
Data1.Age50 <- subset(Data.Mart1, Age=="50")
Data1.Age55 <- subset(Data.Mart1, Age=="55")
Data1.Age56 <- subset(Data.Mart1, Age=="56")
```

```
Data.Mart0 <- subset(Data.clean, Marital_Status=="0")
Data0.Age17 <- subset(Data.Mart0, Age=="17")
Data0.Age25 <- subset(Data.Mart0, Age=="25")
Data0.Age35 <- subset(Data.Mart0, Age=="35")
Data0.Age45 <- subset(Data.Mart0, Age=="45")
Data0.Age50 <- subset(Data.Mart0, Age=="50")
Data0.Age55 <- subset(Data.Mart0, Age=="55")
Data0.Age56 <- subset(Data.Mart0, Age=="56")
```


Pairwise plotting of Marital Status 1 with Age 45:

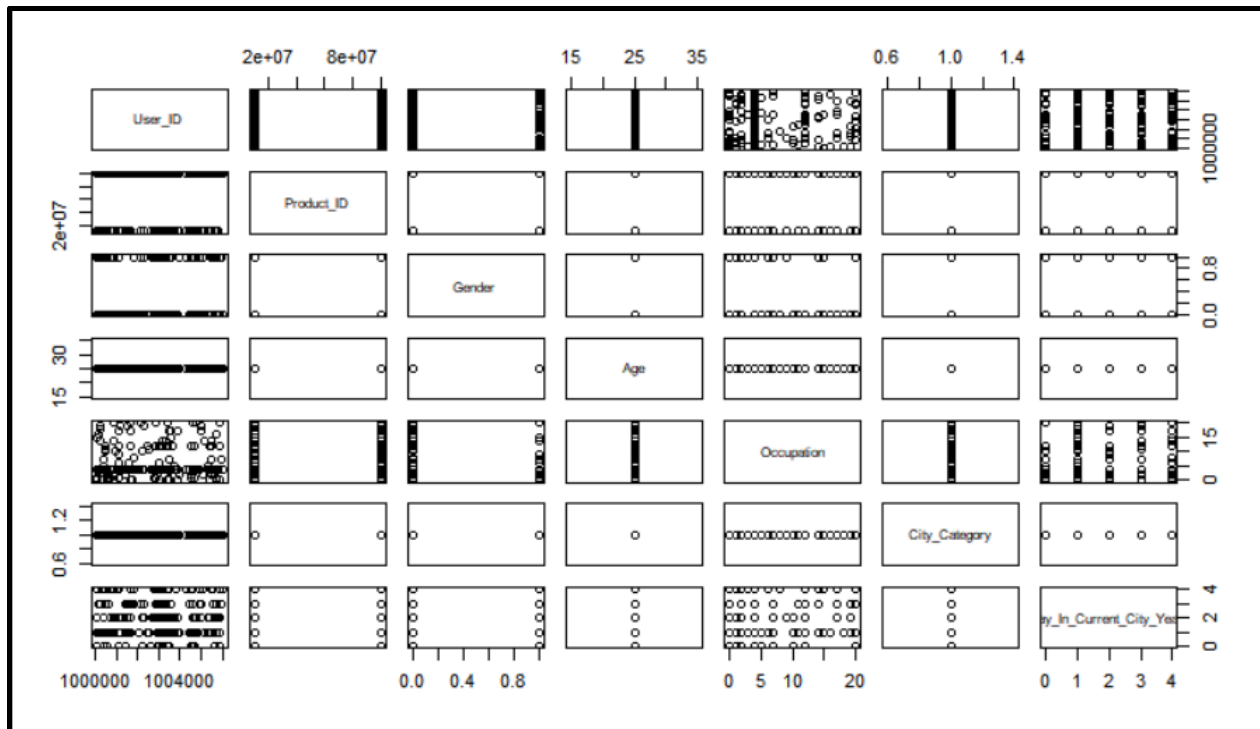
It was interesting to observe behavior of all the features in relation to each other, it was proven to be rather hard to derive any clear correlations between them.

Subsets of City by Category and then by Age:

```
#subset of City by category and then by age
Data.CityA <- subset(Data.clean, City_Category == "1")
Data.A.Age17 <- subset(Data.CityA, Age=="17")
Data.A.Age25 <- subset(Data.CityA, Age=="25")
Data.A.Age35 <- subset(Data.CityA, Age=="35")
Data.A.Age45 <- subset(Data.CityA, Age=="45")
Data.A.Age50 <- subset(Data.CityA, Age=="50")
Data.A.Age55 <- subset(Data.CityA, Age=="55")
Data.A.Age56 <- subset(Data.CityA, Age=="56")
```

```
Data.CityB <- subset(Data.clean, City_Category == "2")
Data.B.Age17 <- subset(Data.CityB, Age=="17")
Data.B.Age25 <- subset(Data.CityB, Age=="25")
Data.B.Age35 <- subset(Data.CityB, Age=="35")
Data.B.Age45 <- subset(Data.CityB, Age=="45")
Data.B.Age50 <- subset(Data.CityB, Age=="50")
Data.B.Age55 <- subset(Data.CityB, Age=="55")
Data.B.Age56 <- subset(Data.CityB, Age=="56")
```

```
Data.CityC <- subset(Data.clean, City_Category == "3")
Data.C.Age17 <- subset(Data.CityC, Age=="17")
Data.C.Age25 <- subset(Data.CityC, Age=="25")
Data.C.Age35 <- subset(Data.CityC, Age=="35")
Data.C.Age45 <- subset(Data.CityC, Age=="45")
Data.C.Age50 <- subset(Data.CityC, Age=="50")
Data.C.Age55 <- subset(Data.CityC, Age=="55")
Data.C.Age56 <- subset(Data.CityC, Age=="56")
```

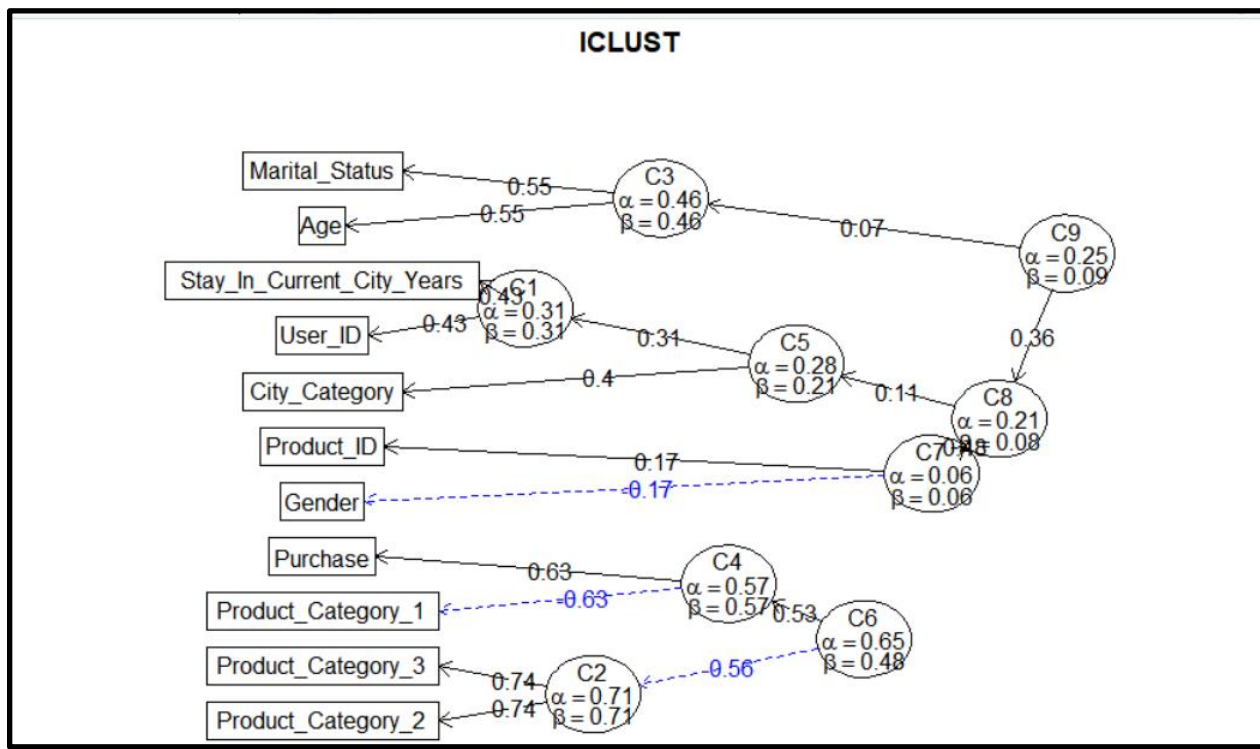
Pairwise plotting for City Category A with Age 25:

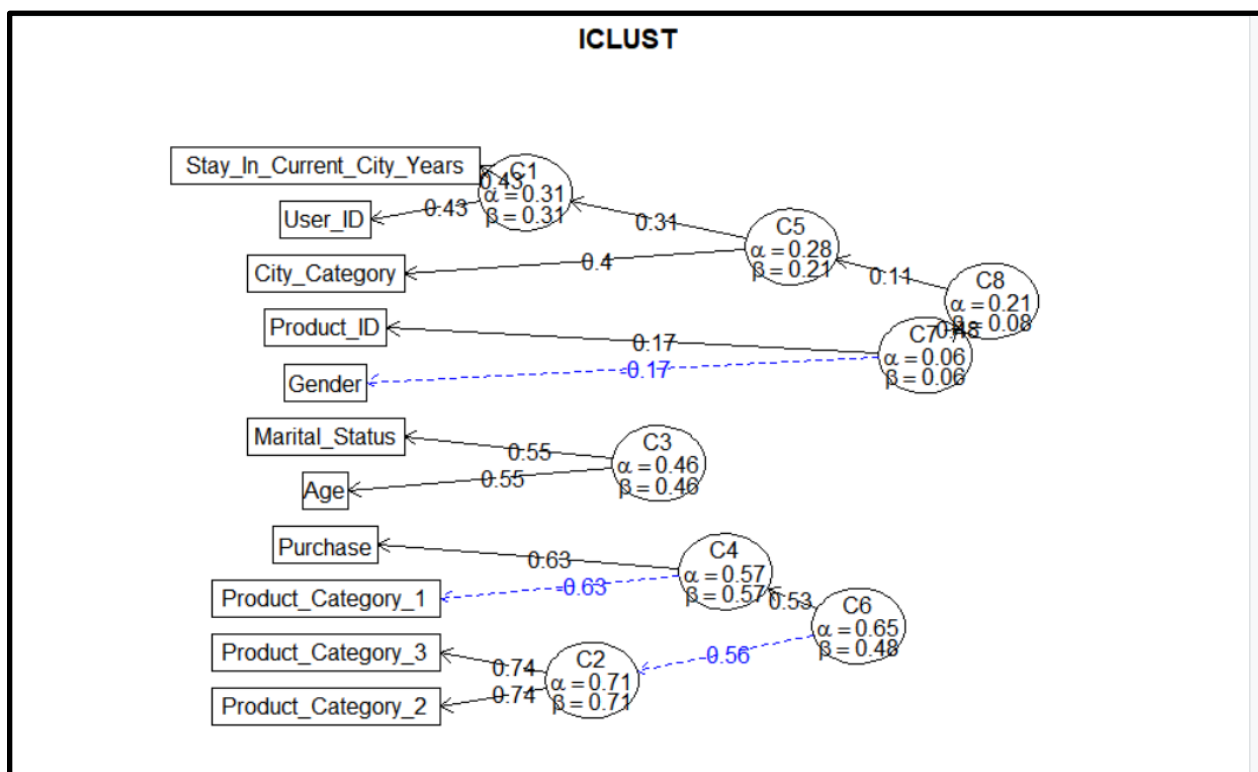
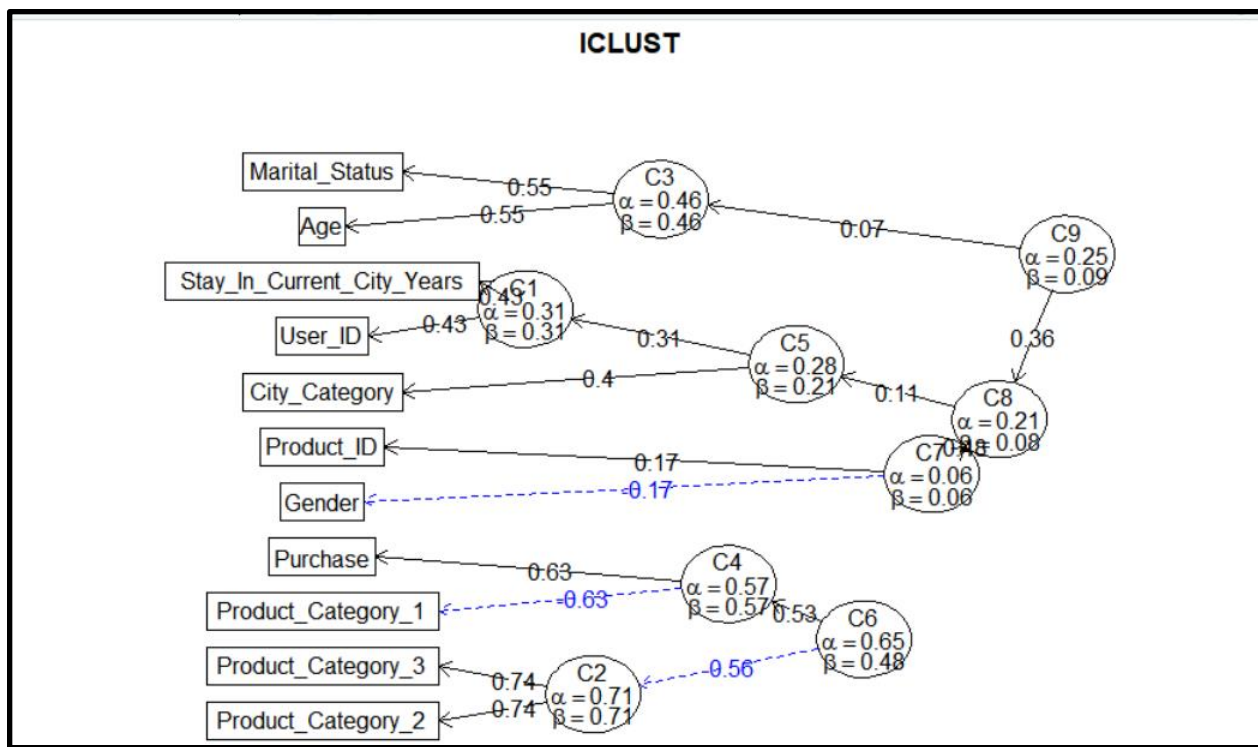
ICLUST**Occupation 0**

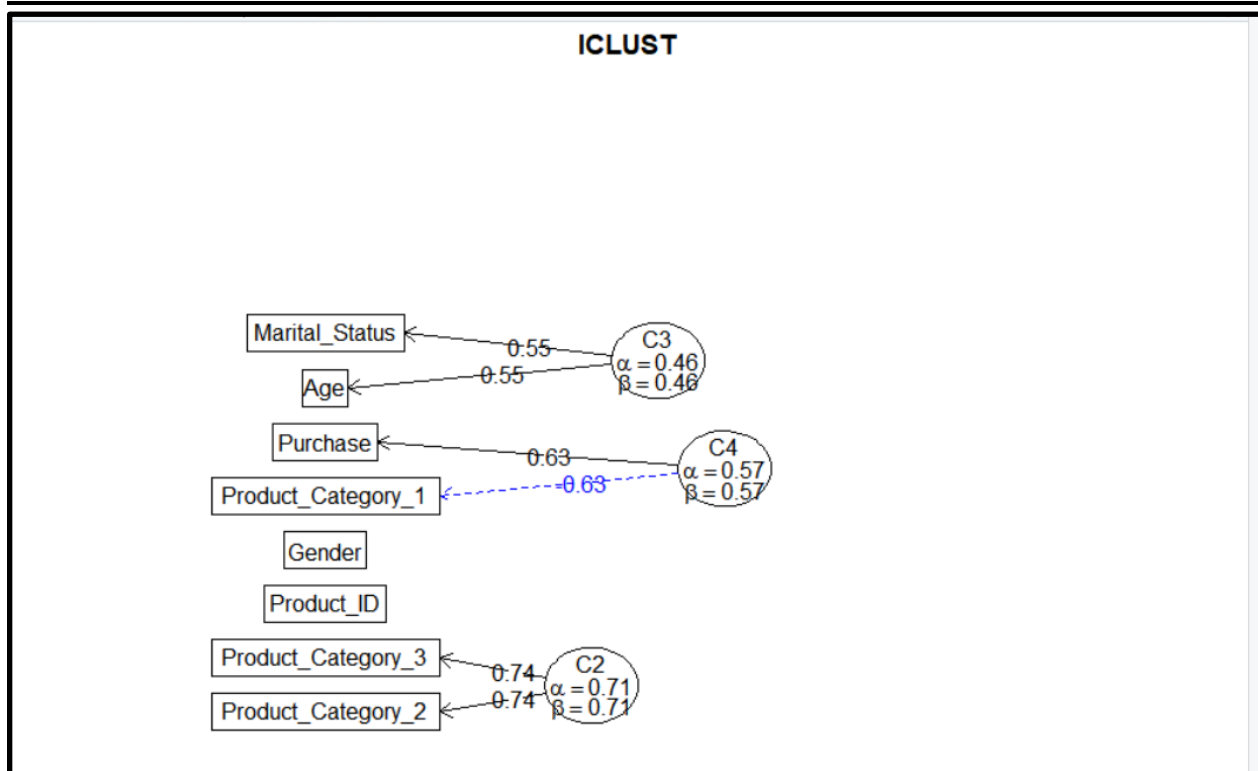
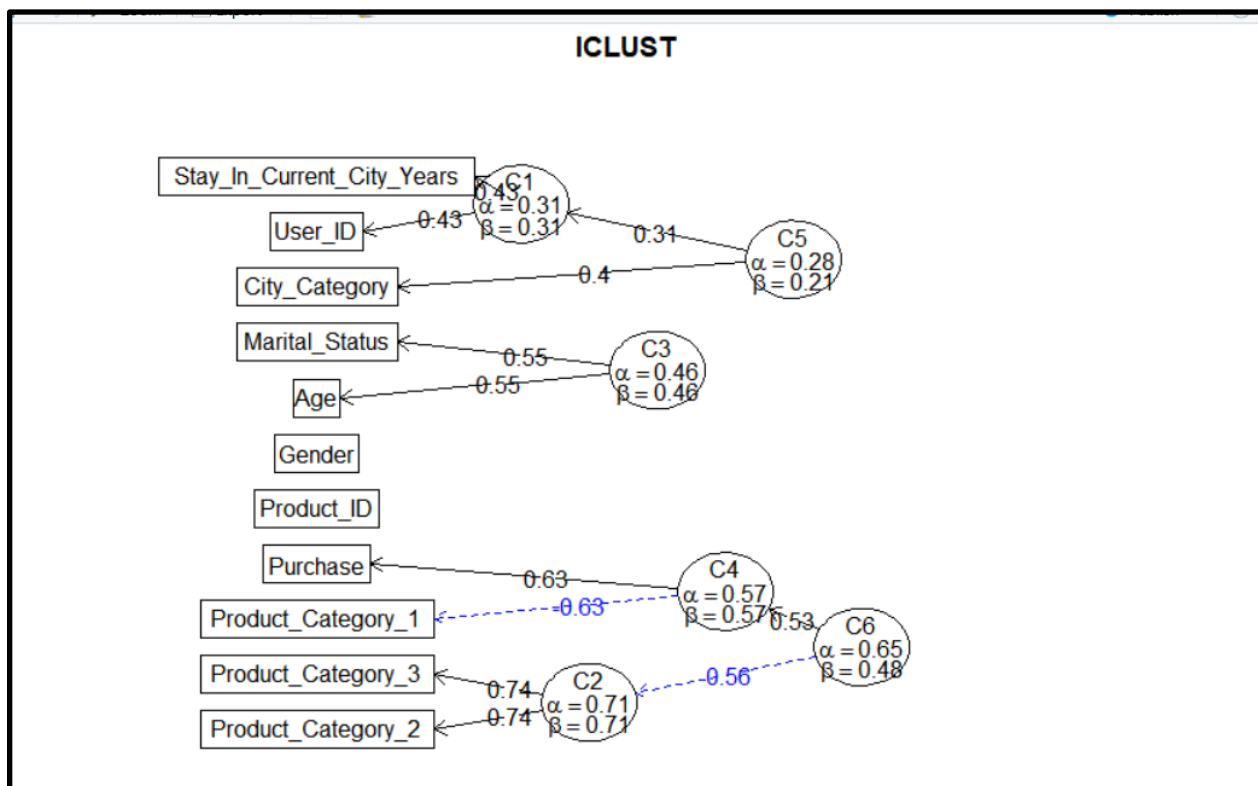
```

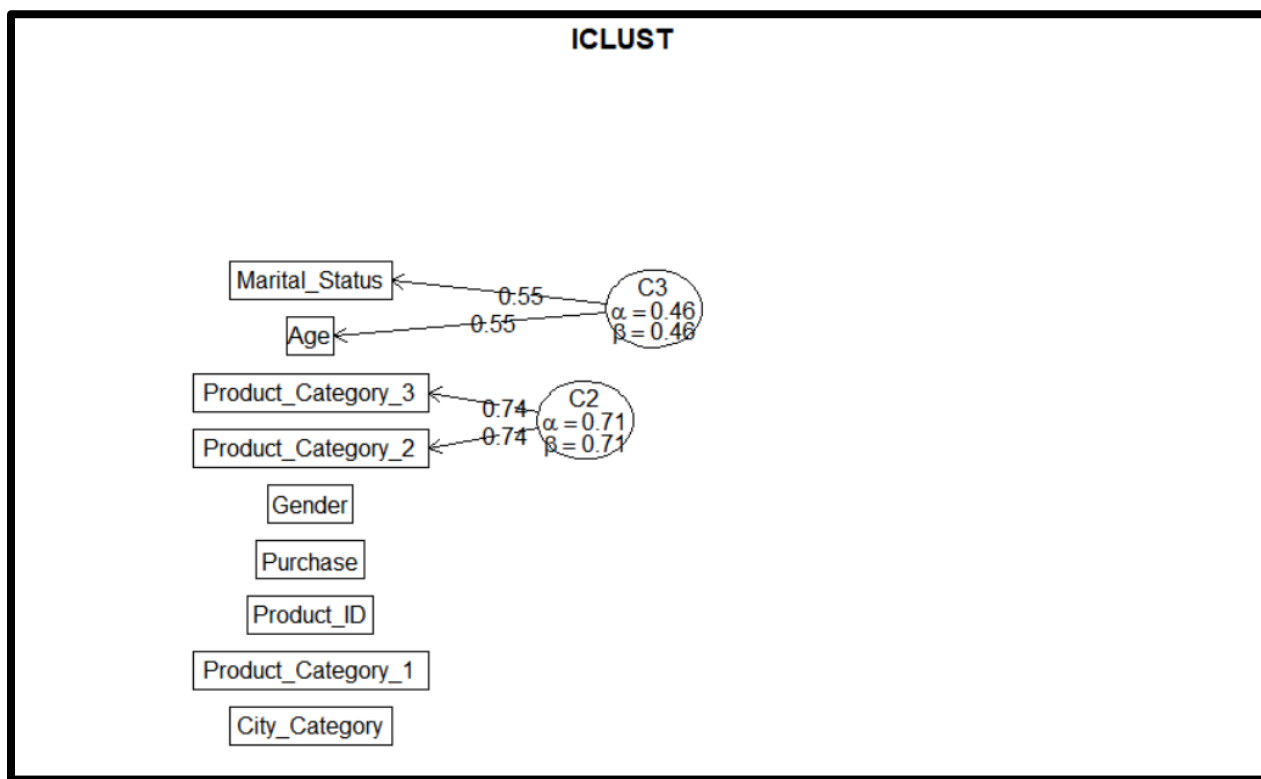
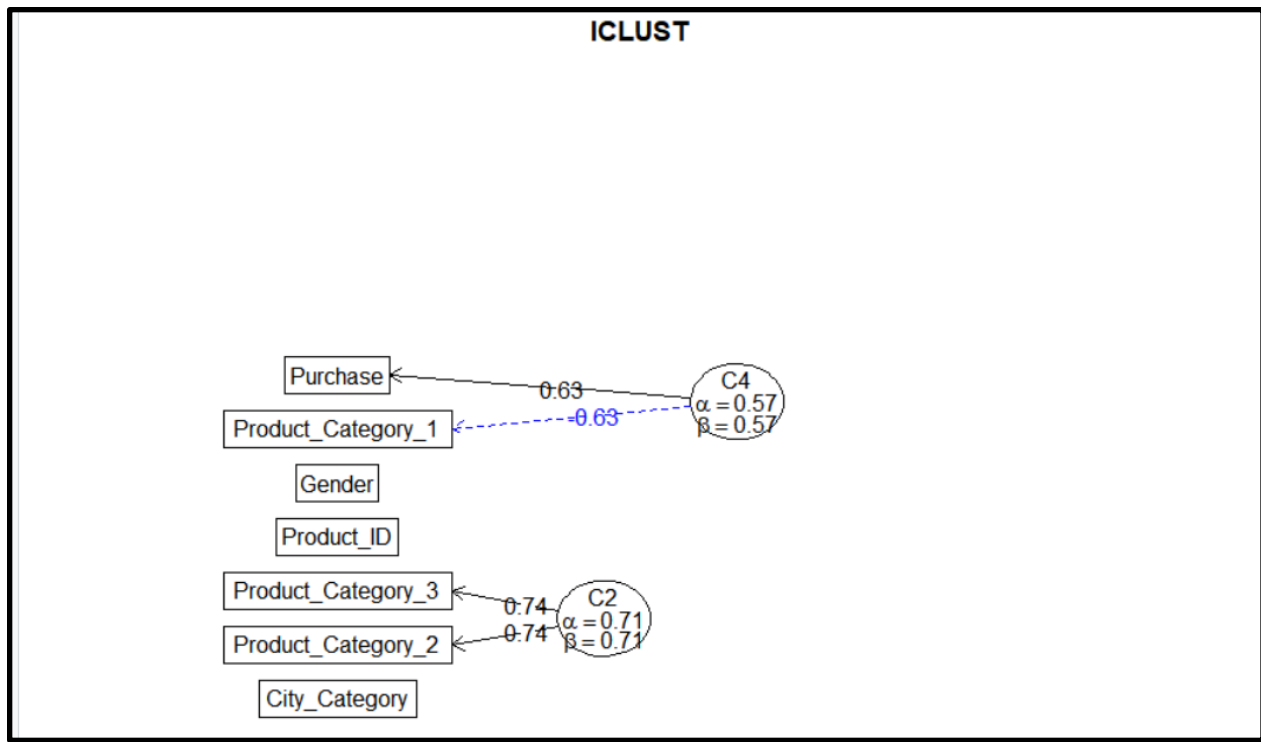
iclust(Data.Occ0,nclusters=2)
iclust(Data.Occ0,nclusters=3)
iclust(Data.Occ0,nclusters=4)
iclust(Data.Occ0,nclusters=5)
iclust(Data.Occ0,nclusters=6)
iclust(Data.Occ0,nclusters=7)
iclust(Data.Occ0,nclusters=8)
iclust(Data.Occ0,nclusters=9)
iclust(Data.Occ0,nclusters=10)

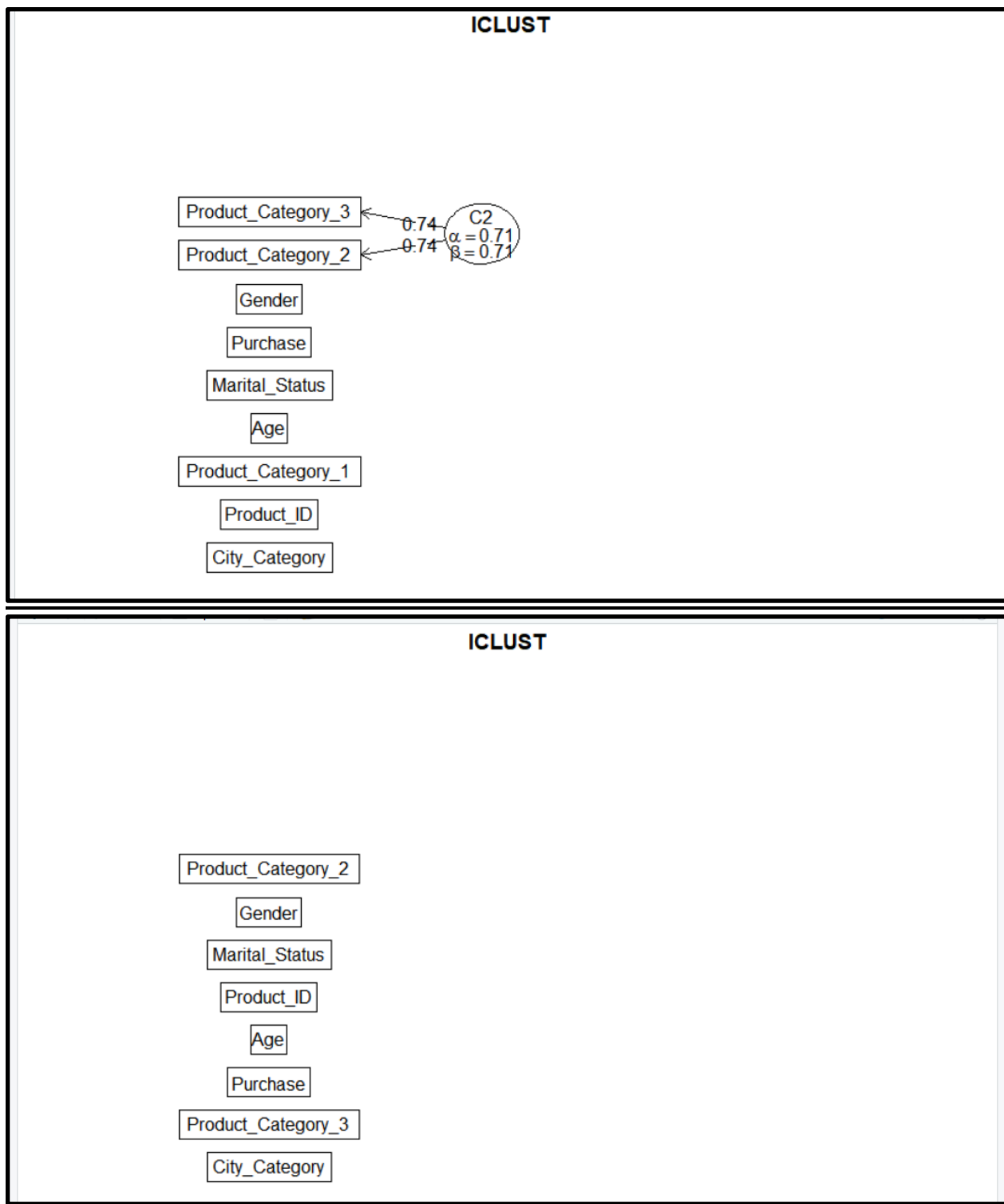
```









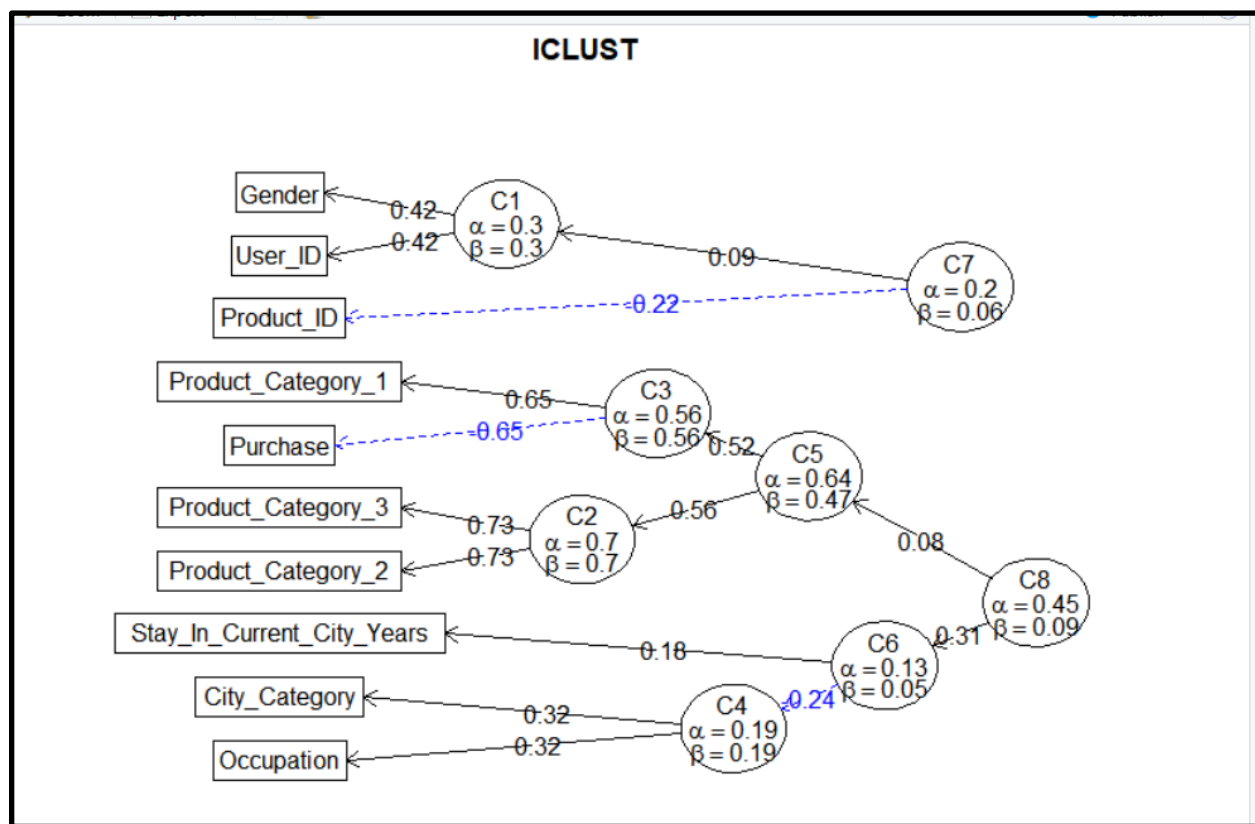


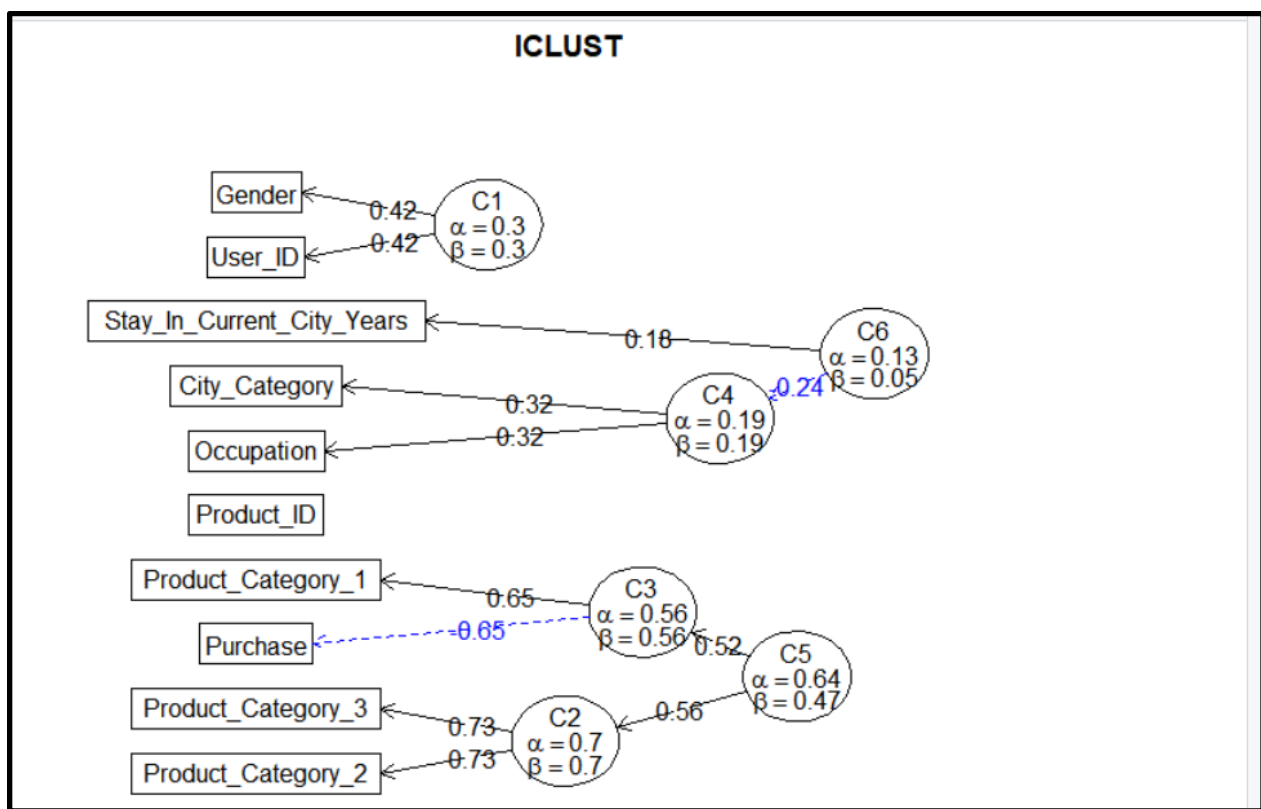
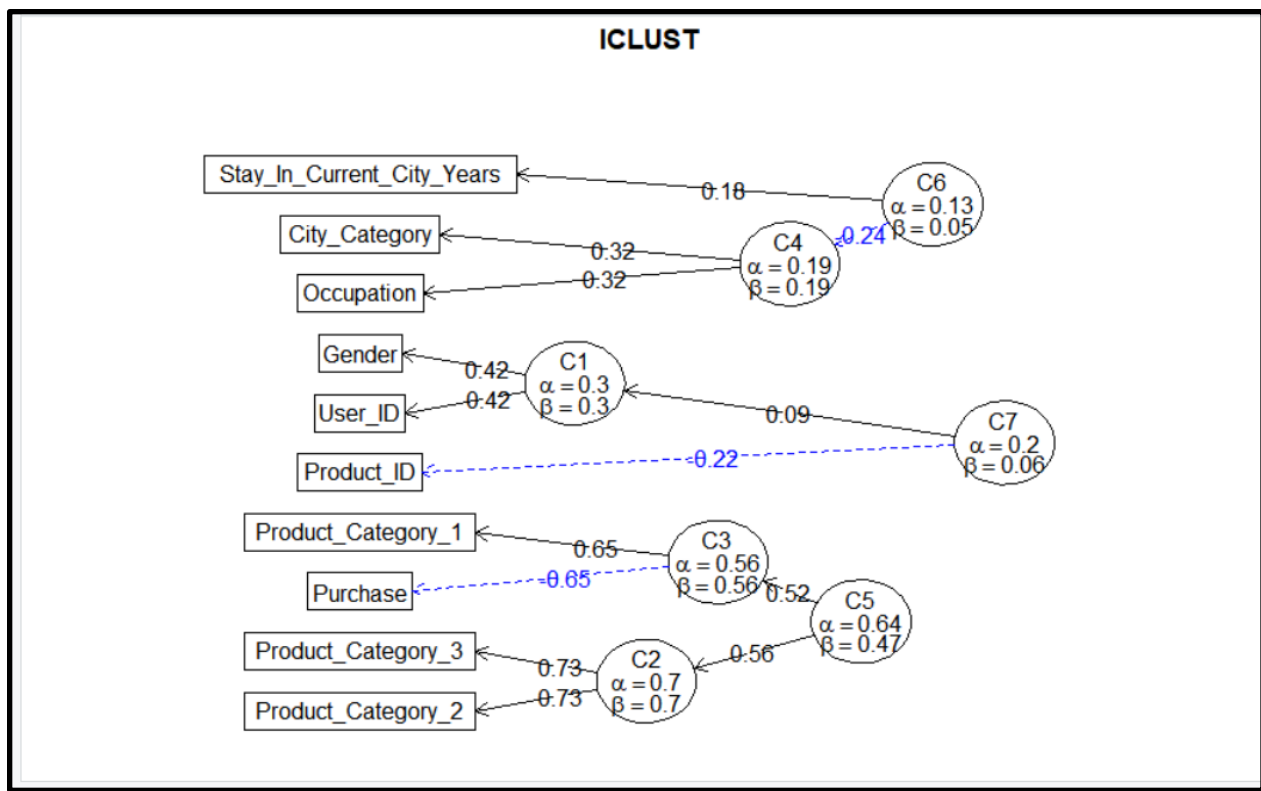
From the Iclust graphs, we can see that, as the number of clusters increases, the number of features being neglected in the process of clustering also increases. We can conclude that $k=4$ will be the most optimal solution with the number of features being the ones which are

important. We can clearly see that purchase and product Id have lesser importance and observe it for the Occupation 0.

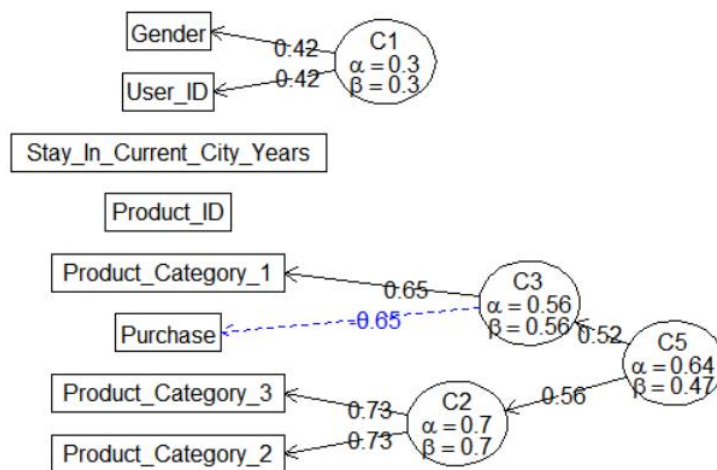
Marital Status 1 with Age 45

```
iclust(Data1.Age45,nclusters=2)
iclust(Data1.Age45,nclusters=3)
iclust(Data1.Age45,nclusters=4)
iclust(Data1.Age45,nclusters=5)
iclust(Data1.Age45,nclusters=6)
iclust(Data1.Age45,nclusters=7)
iclust(Data1.Age45,nclusters=8)
iclust(Data1.Age45,nclusters=9)
iclust(Data1.Age45,nclusters=10)
```

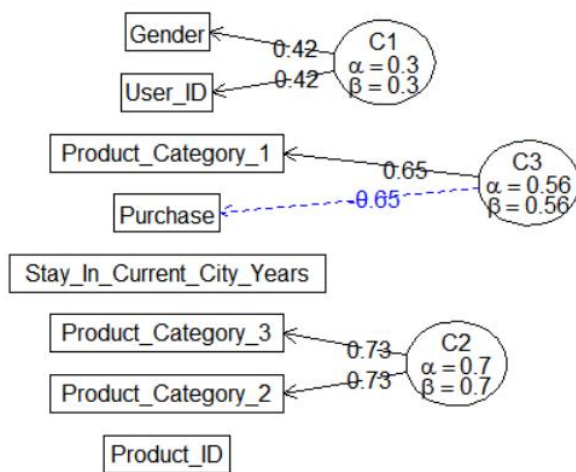


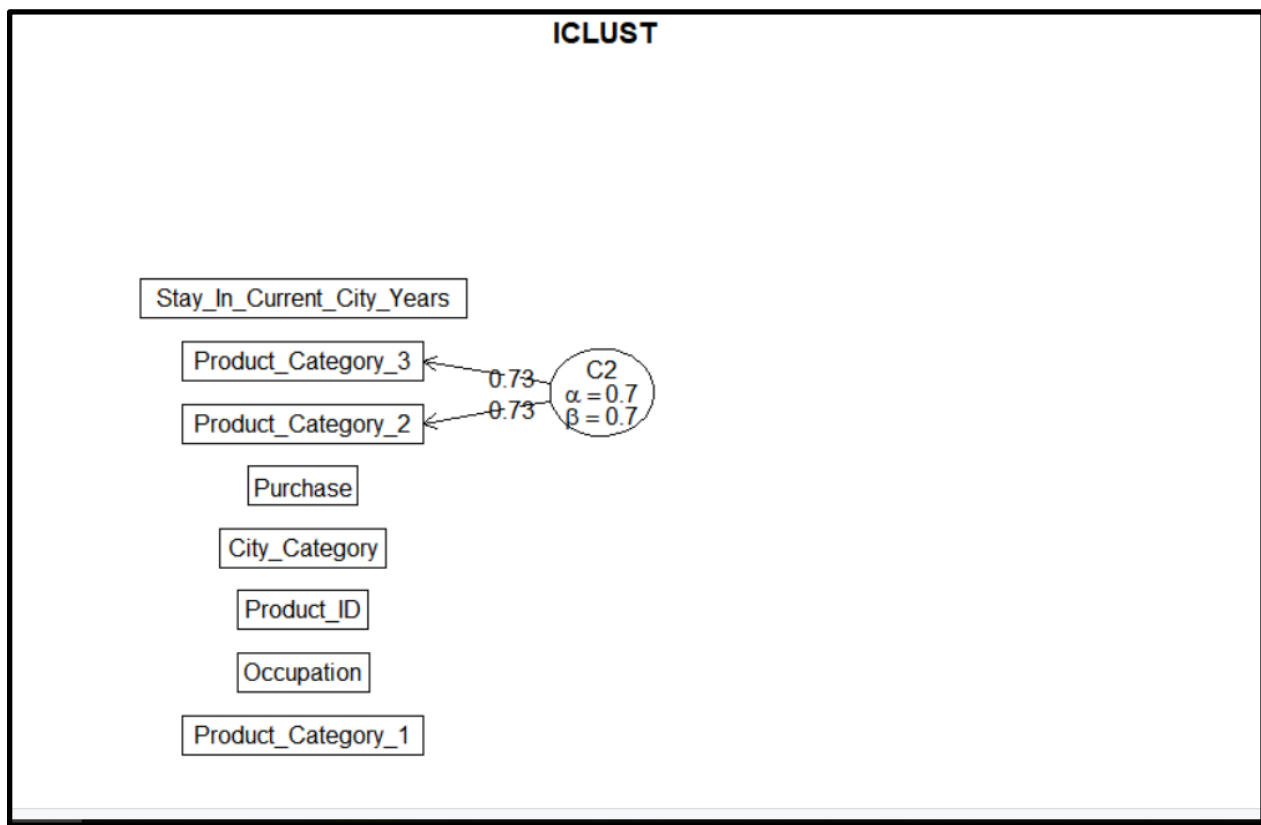
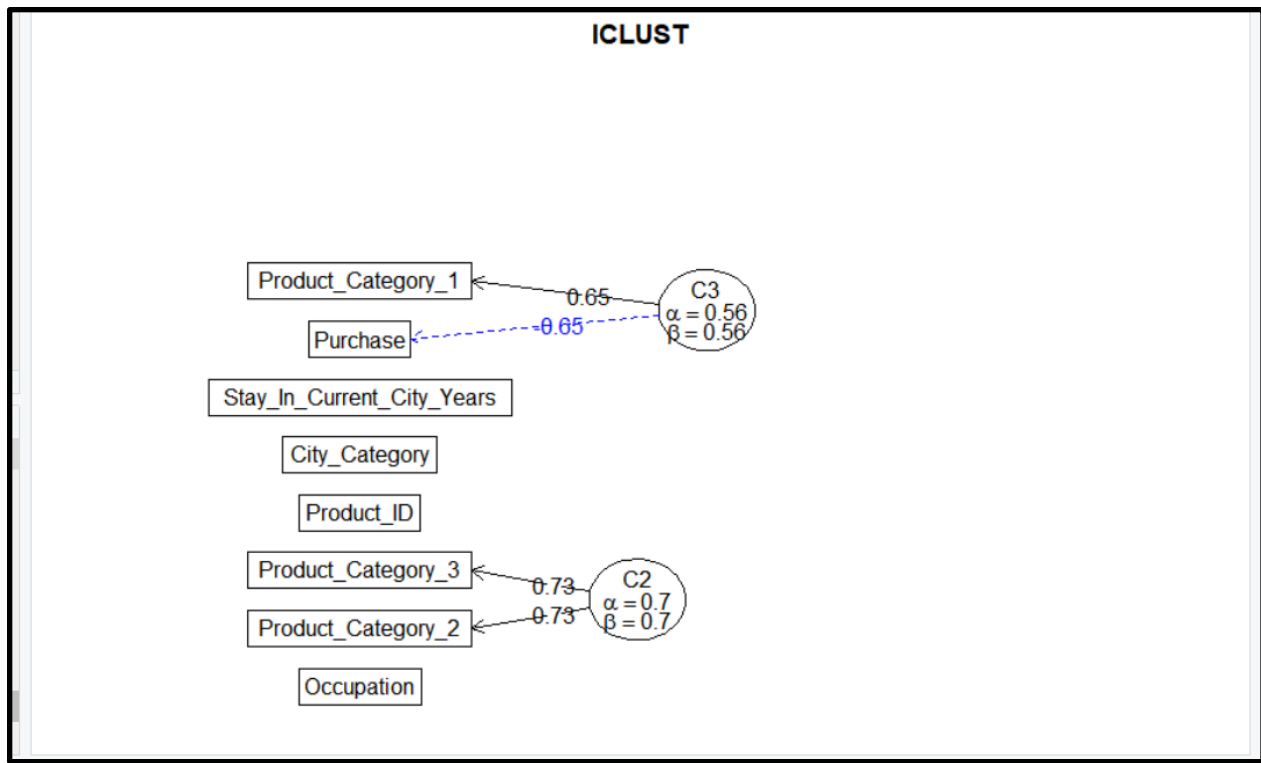


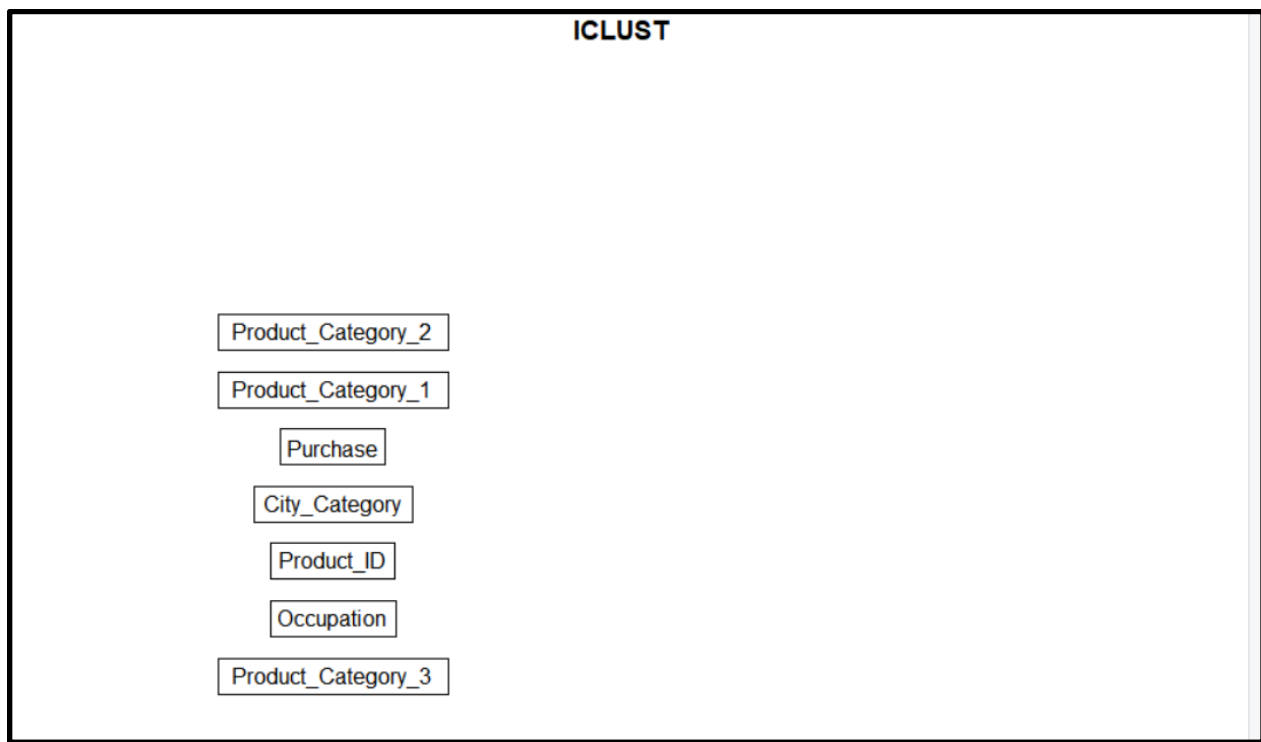
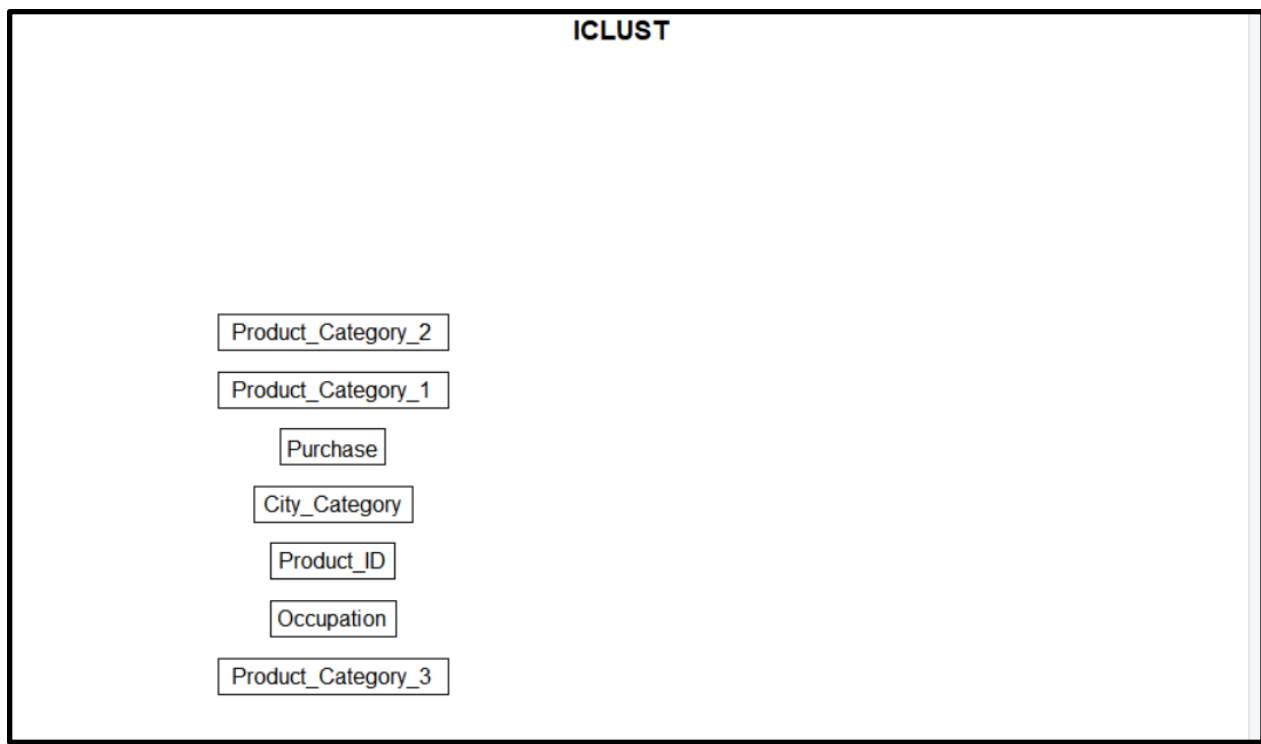
ICLUST



ICLUST





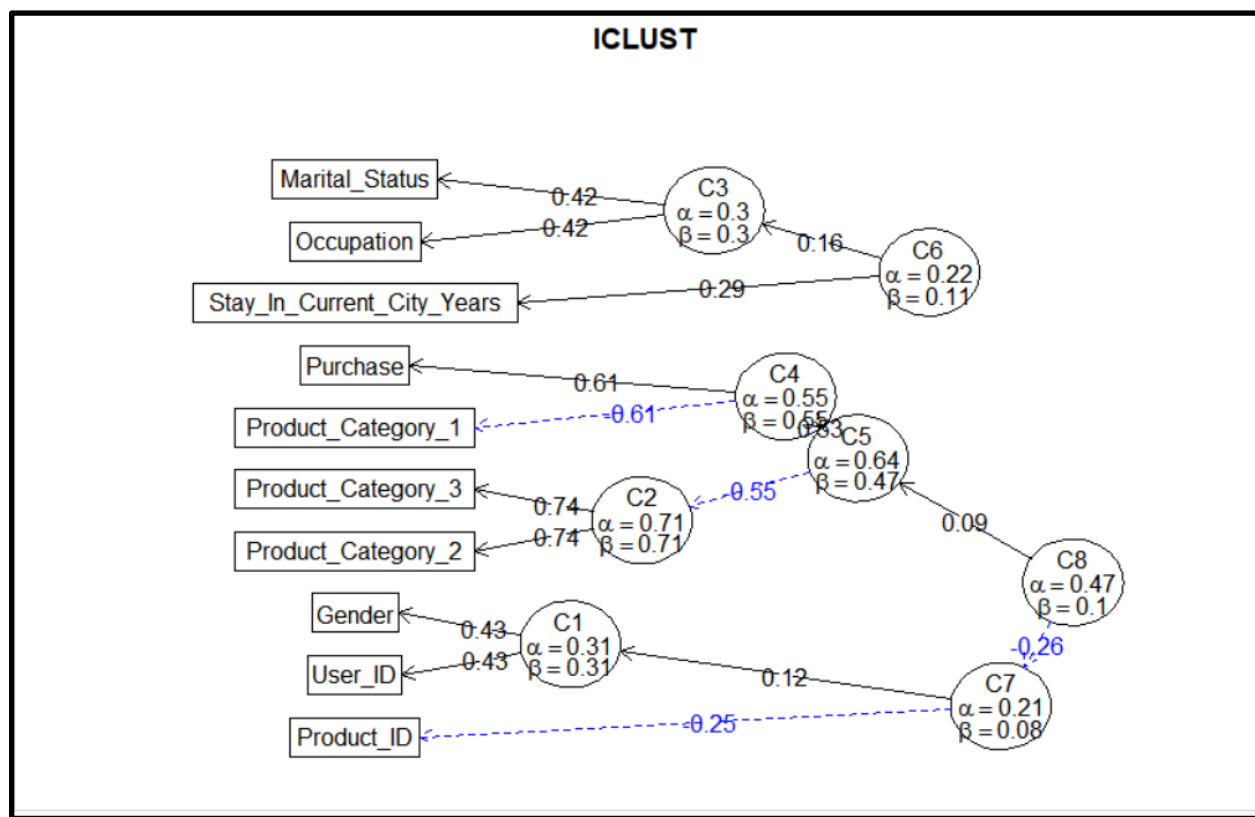


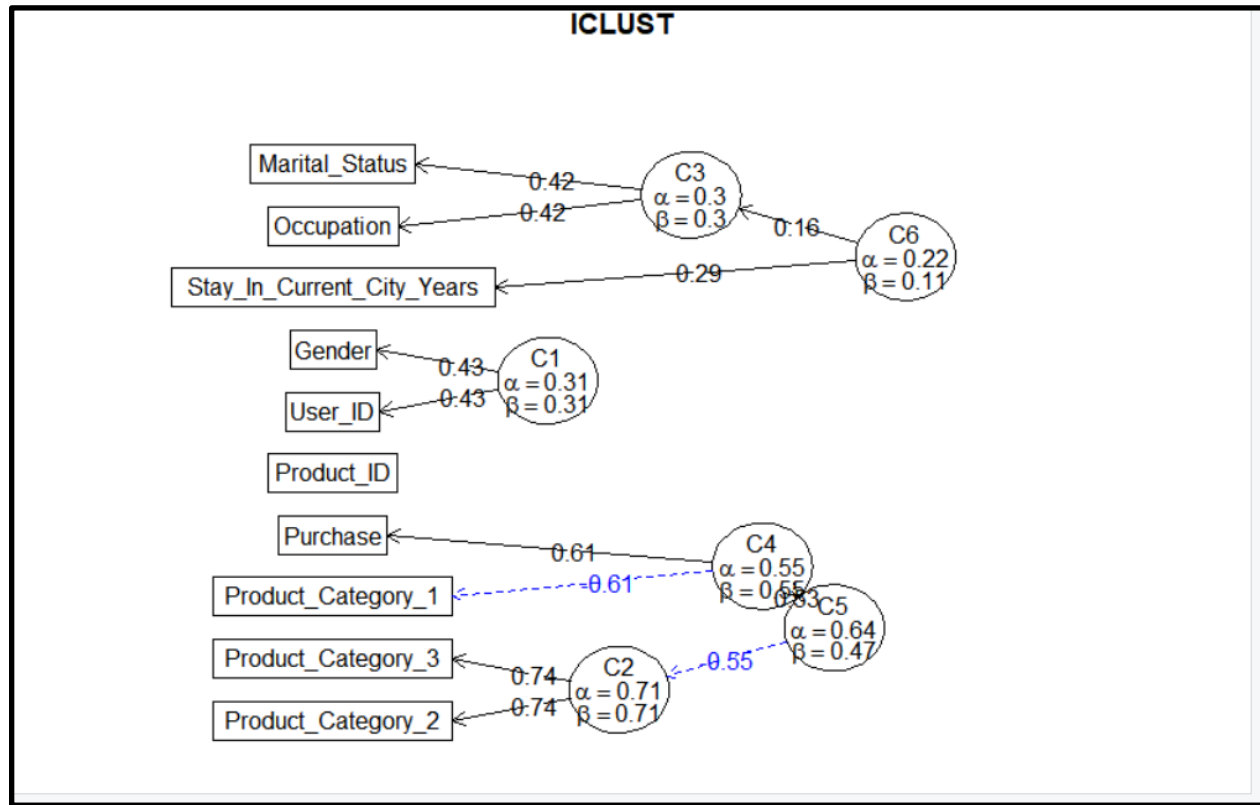
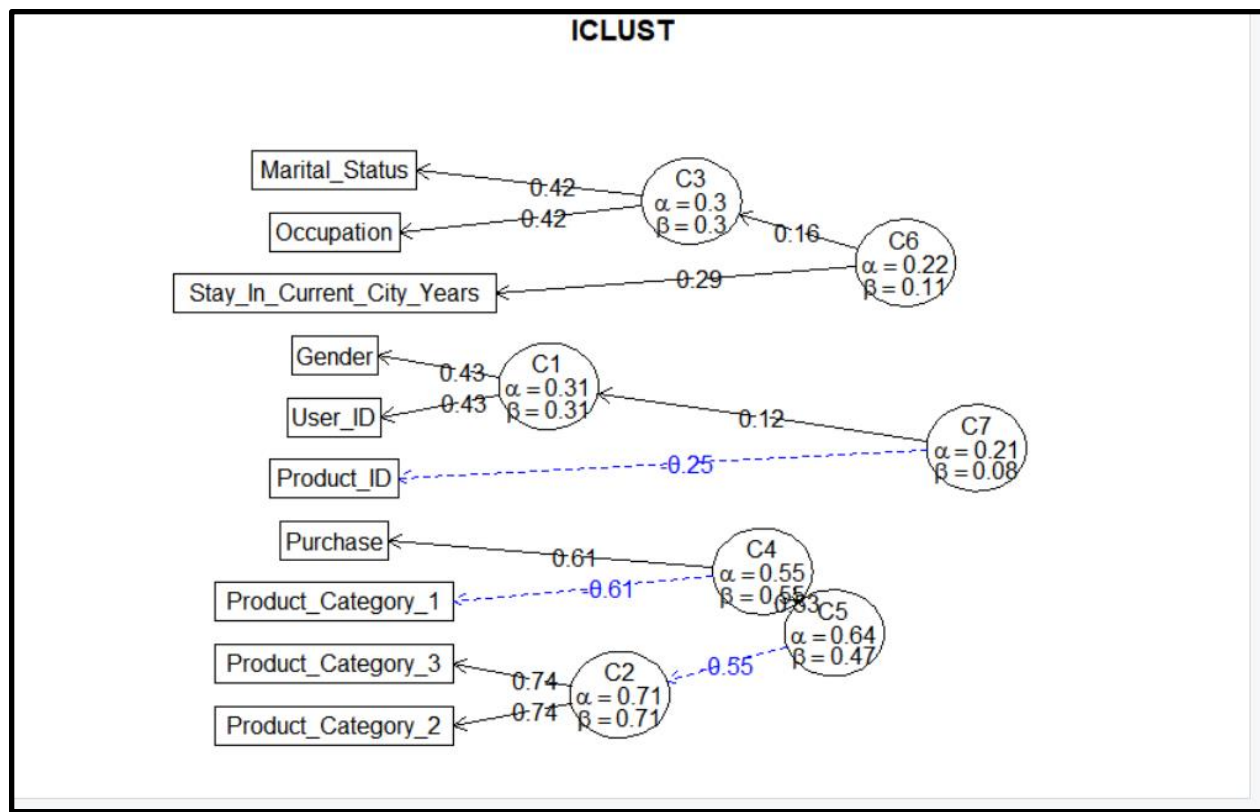
From the Iclust graphs, we can see that, as the number of clusters increases, the number of features being neglected in the process of clustering also increases. We can conclude that $k=4$ will be the most optimal solution with the number of features being the ones which are

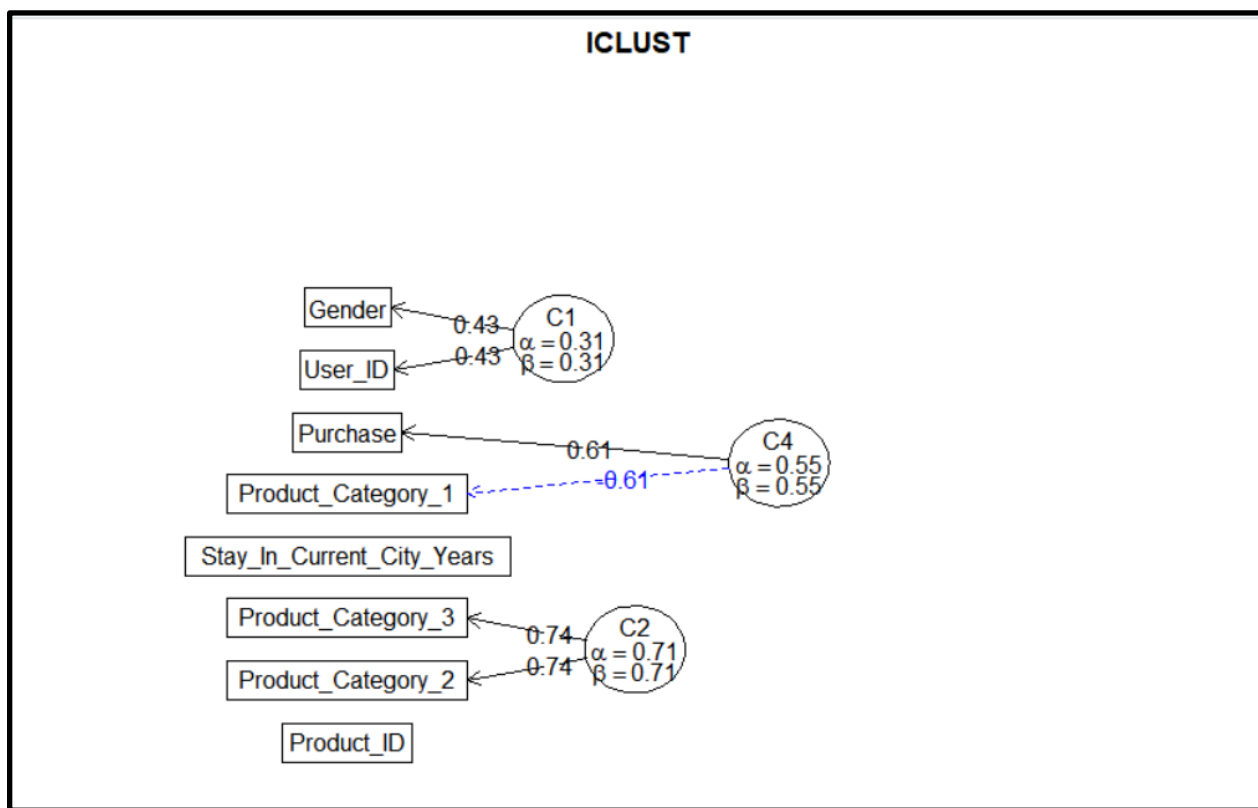
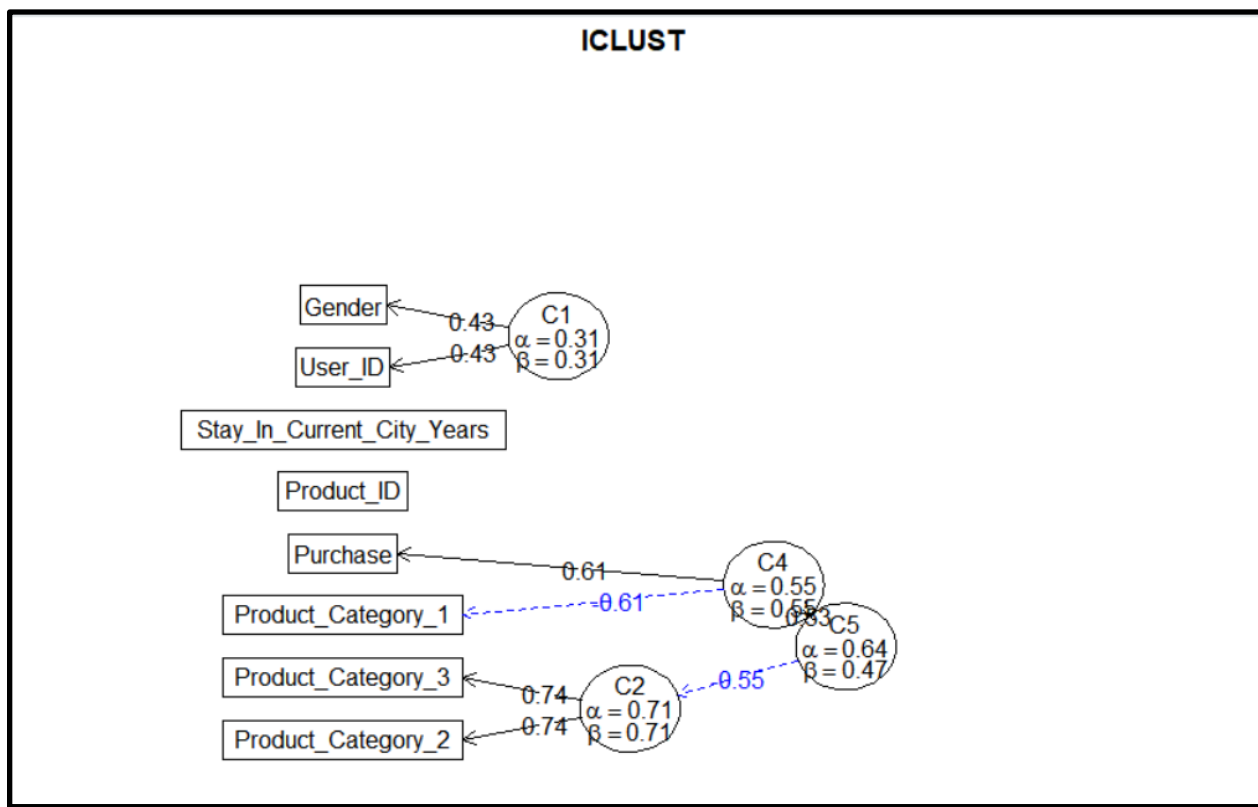
important. We can clearly see that stay in current city years and product Id have lesser importance and observe it for Marital Status 1 with Age 45.

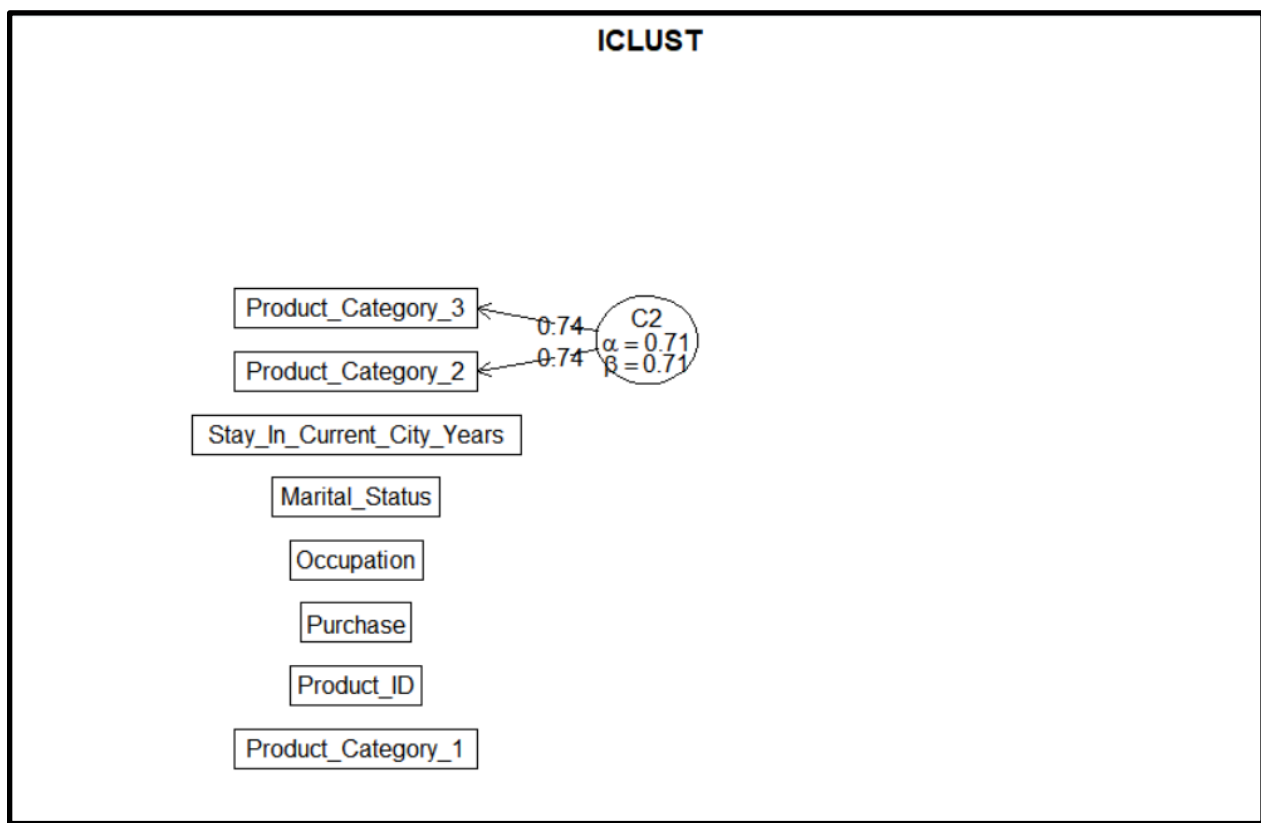
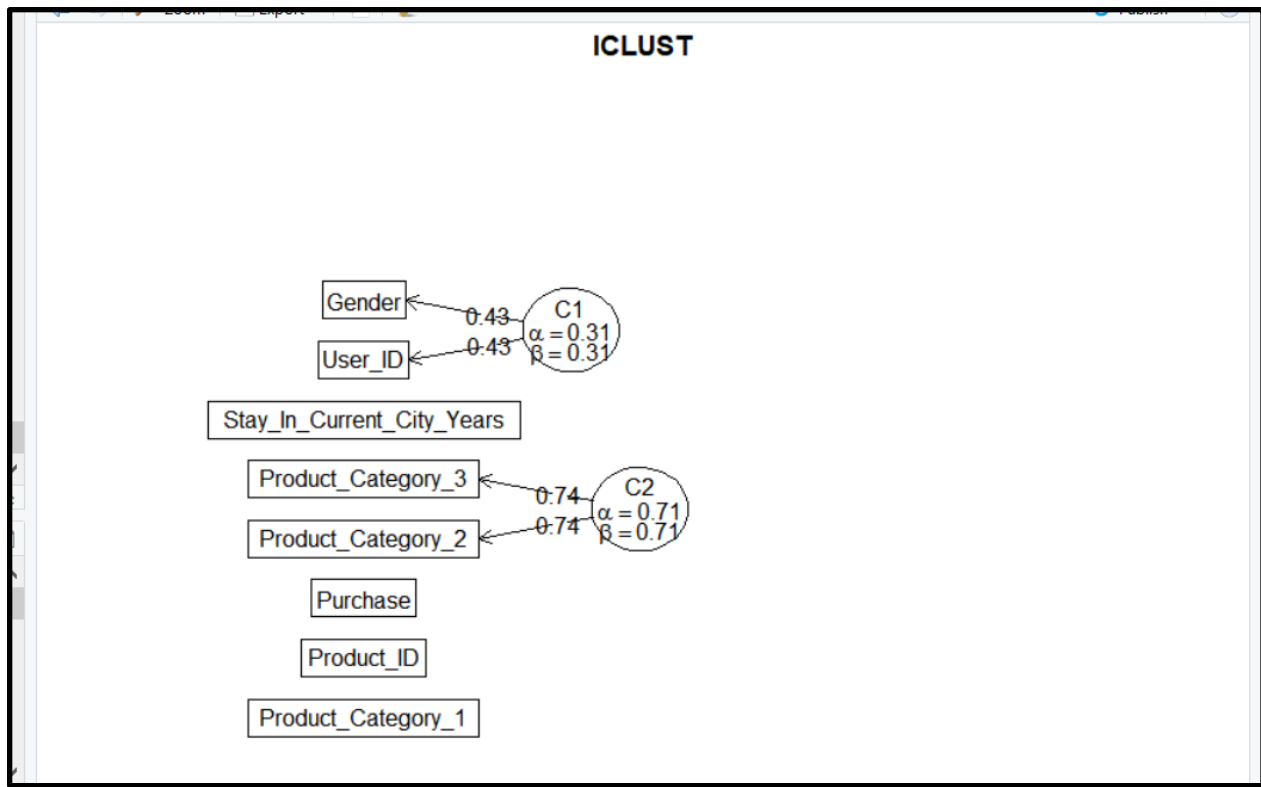
City Category A with Age 25

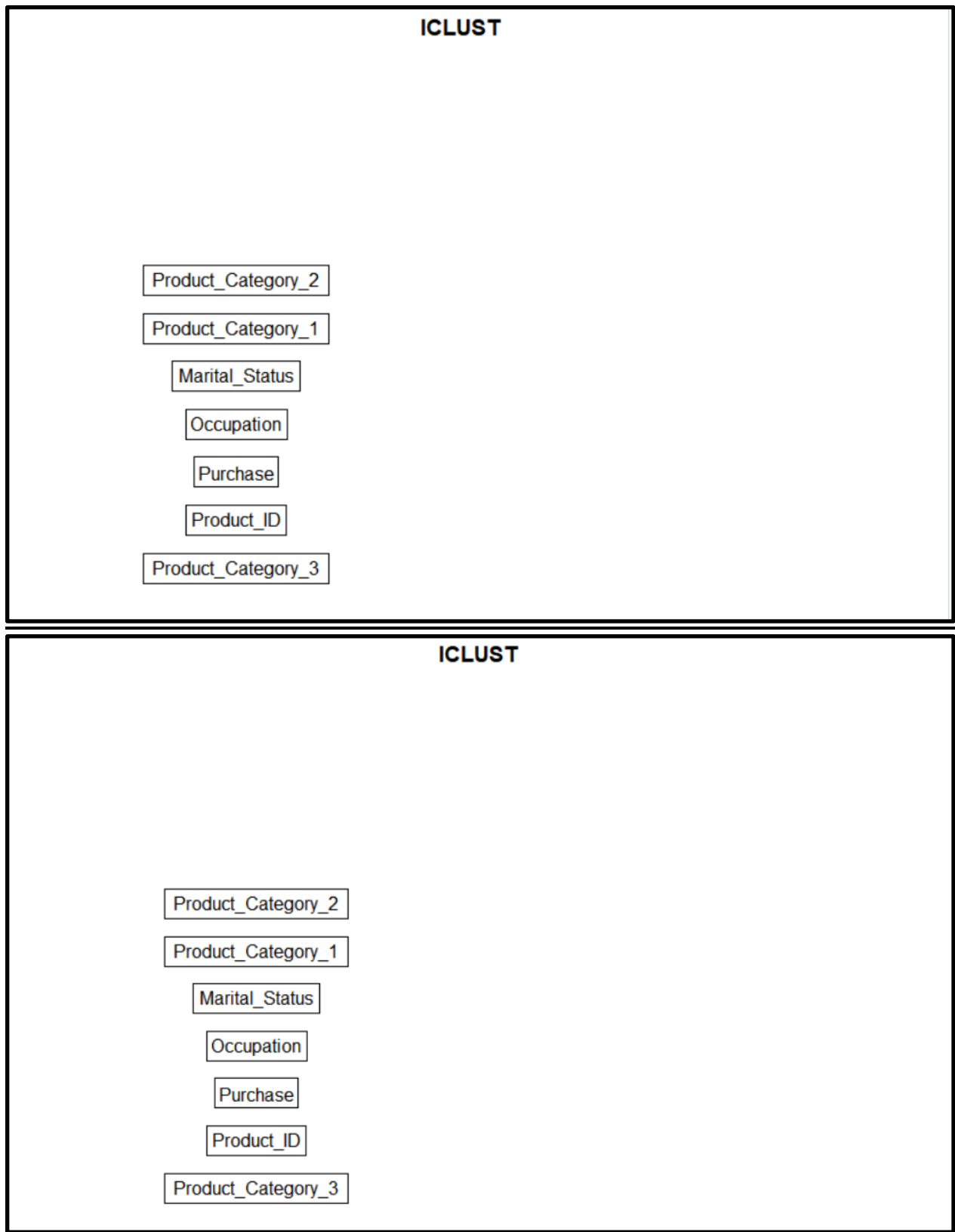
```
iclust(Data.A.Age25,nclusters=2)
iclust(Data.A.Age25,nclusters=3)
iclust(Data.A.Age25,nclusters=4)
iclust(Data.A.Age25,nclusters=5)
iclust(Data.A.Age25,nclusters=6)
iclust(Data.A.Age25,nclusters=7)
iclust(Data.A.Age25,nclusters=8)
iclust(Data.A.Age25,nclusters=9)
iclust(Data.A.Age25,nclusters=10)
```











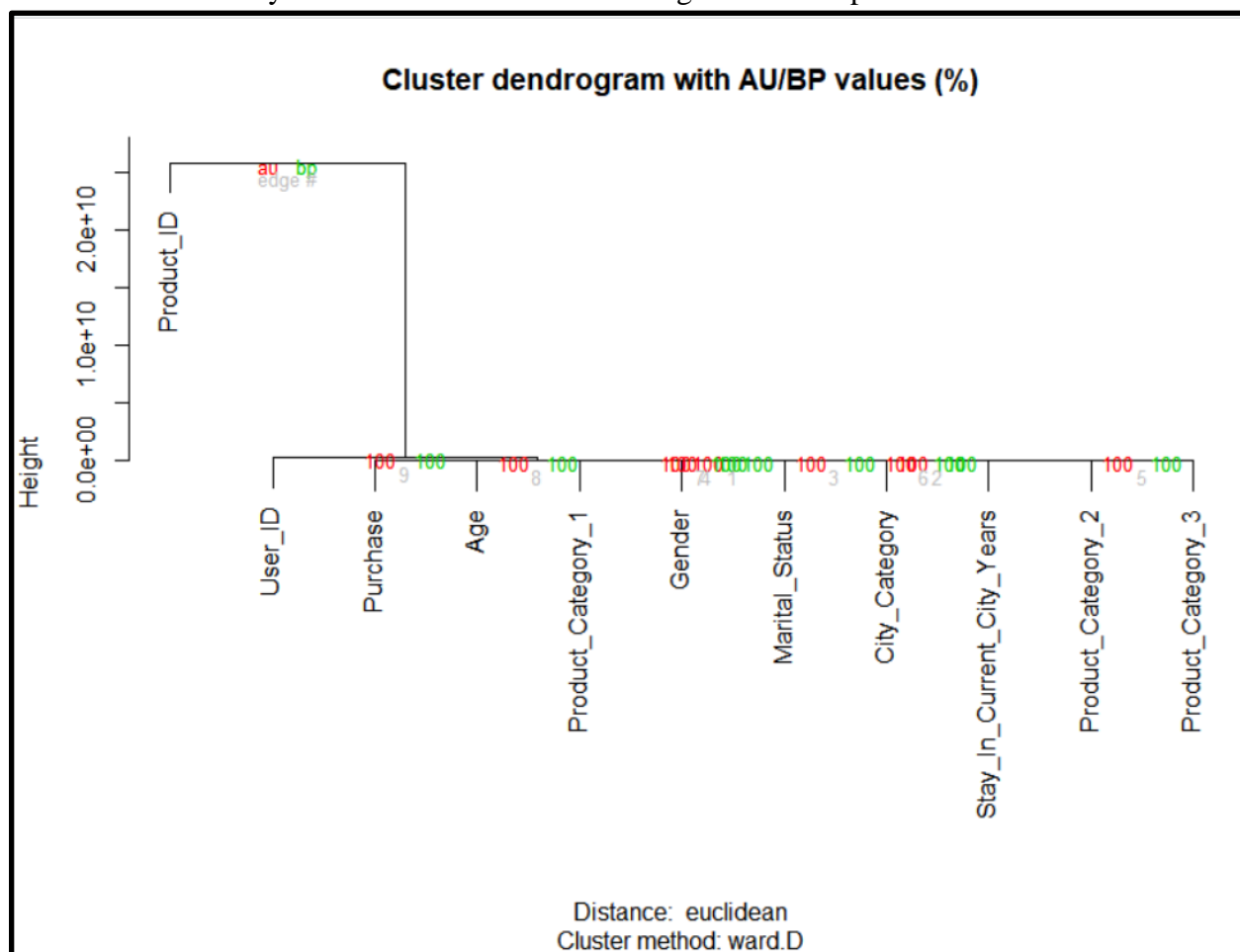
From the Iclust graphs, we can see that, as the number of clusters increases, the number of features being neglected in the process of clustering also increases. We can conclude that $k=4$ will be the most optimal solution with the number of features being the ones which are important. We can clearly see that stay in current city years and product Id have lesser importance and observe it for City Category A with Age 25.

Ward Hierarchical Clustering

A different method of hierarchical clustering and compare it to iClust, the method of Ward Hierarchical Clustering was used. The following function were used:

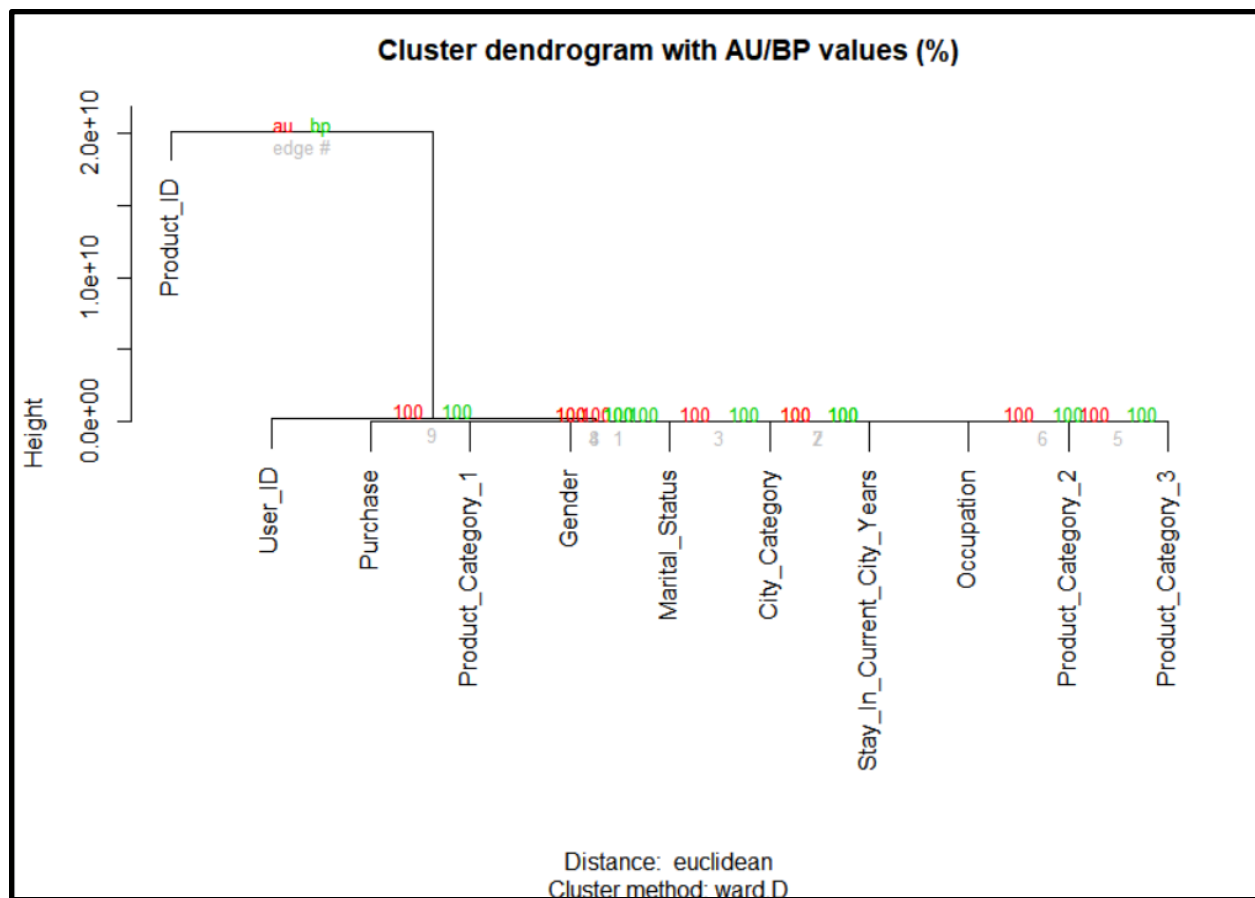
```
fit <- pvclust(Data.Occ0, method.hclust="ward", method.dist="euclidean")
plot(fit)
```

This method of clustering does not accept a given number of clusters. Instead, it chooses the number of clusters by itself. The below is the dendrogram of Occupation 0.



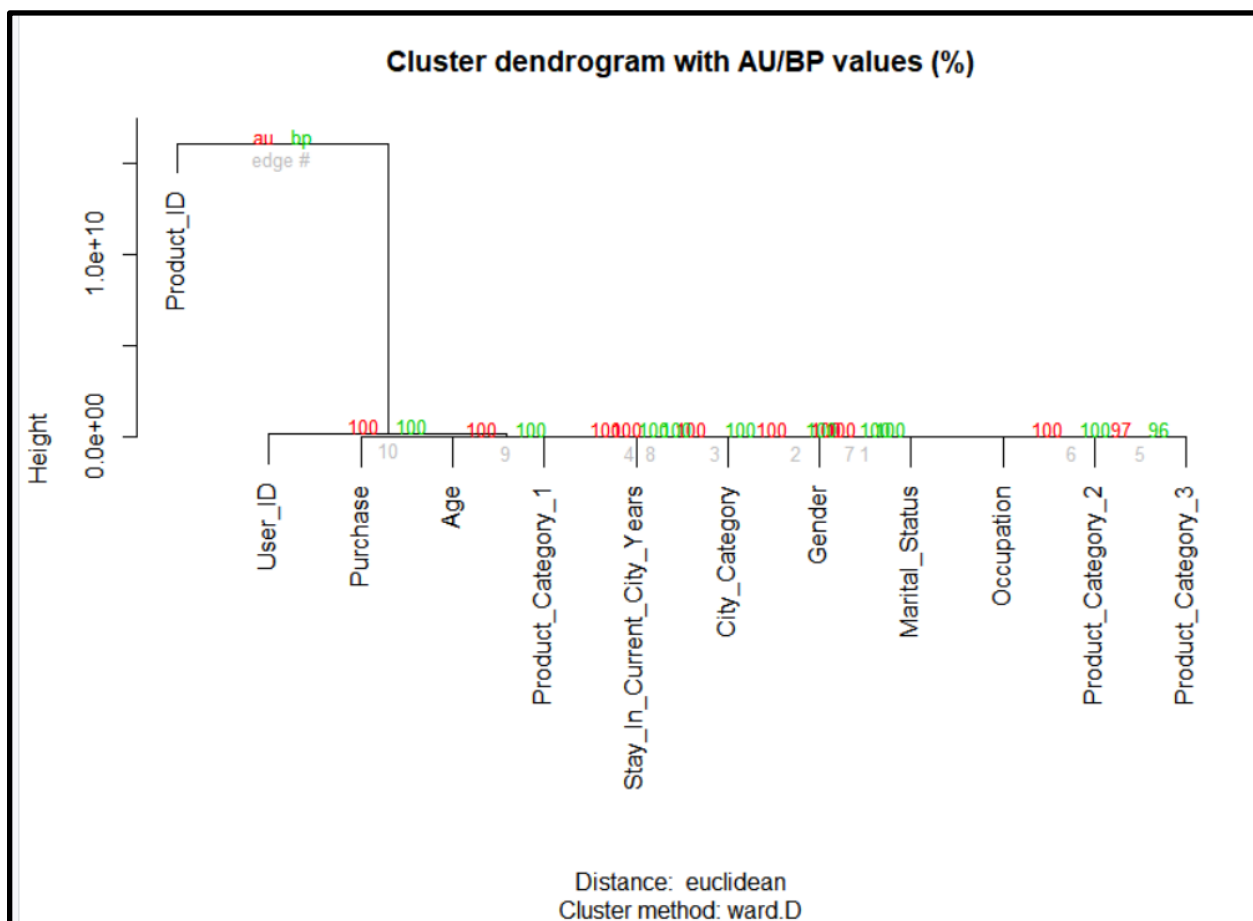
```
fit <- pvcust(Data.Age45, method.hclust="ward", method.dist="euclidean")  
plot(fit)
```

The below is the dendrogram for Marital Status 1 with Age 45.



```
fit <- pvclust(Data.A.Age25, method.hclust="ward", method.dist="euclidean")
plot(fit)
```

The below is the dendrogram for City Category A with age 25.



Clustering

Occupation 0

Kmeans

```
#K Means#  
#OCCUPATION 0#  
#K means for cluster N = 2  
km.Occ0.2 = kmeans(Data.Occ0,2,nstart = 25,iter.max = 15)  
#K means for cluster N = 3  
km.Occ0.3 = kmeans(Data.Occ0,3,nstart = 25,iter.max = 15)  
#K means for cluster N = 4  
km.Occ0.4 = kmeans(Data.Occ0,4,nstart = 25,iter.max = 15)  
#K means for cluster N = 5  
km.Occ0.5 = kmeans(Data.Occ0,5,nstart = 25,iter.max = 15)  
#K means for cluster N = 6  
km.Occ0.6 = kmeans(Data.Occ0,6,nstart = 25,iter.max = 15)  
#K means for cluster N = 7  
km.Occ0.7 = kmeans(Data.Occ0,7,nstart = 25,iter.max = 15)  
#K means for cluster N = 8  
km.Occ0.8 = kmeans(Data.Occ0,8,nstart = 25,iter.max = 15)  
#K means for cluster N = 9  
km.Occ0.9 = kmeans(Data.Occ0,9,nstart = 25,iter.max = 15)  
#K means for cluster N = 10  
km.Occ0.10 = kmeans(Data.Occ0,10,nstart = 25,iter.max = 15)
```

K-median

```
#loading Gmedian library  
install.packages("Gmedian")  
library(Gmedian)
```

```
Occ0 <- as.matrix(Data.Occ0)
```

```
#K-median for 2-10 Clusters  
kmed.Occ0.2 <- kGmedian(Occ0, ncenters = 2)  
kmed.Occ0.3 <- kGmedian(Occ0, ncenters = 3)  
kmed.Occ0.4 <- kGmedian(Occ0, ncenters = 4)  
kmed.Occ0.5 <- kGmedian(Occ0, ncenters = 5)  
kmed.Occ0.6 <- kGmedian(Occ0, ncenters = 6)  
kmed.Occ0.7 <- kGmedian(Occ0, ncenters = 7)  
kmed.Occ0.8 <- kGmedian(Occ0, ncenters = 8)  
kmed.Occ0.9 <- kGmedian(Occ0, ncenters = 9)  
kmed.Occ0.10 <- kGmedian(Occ0, ncenters = 10)
```

KNN

```

set.seed(1234)
data_norm<-function(x){((x-min(x)/ (max(x)-min(x))))}
Data.Occ0_Norm<-as.data.frame(lapply(Data.Occ0, data_norm))
ind<-sample(2, nrow(Data.Occ0_Norm), replace=TRUE, prob = c(0.7,0.3))
train.data<-Data.Occ0_Norm[ind==1,]
test.data<-Data.Occ0_Norm[ind==2,]
predict.data<-knn(train.data, test.data, train.data[,6], k =50 , prob=TRUE)
table(predict.data,test.data[,6])

```

Marital Status 1 with Age 45:**Kmeans**

```

#K Means#
#Marital Status 1#
#K Means#
#Marital Status 1#
#AGE 45#
#K means for cluster N = 2
km.Data1.Age45 = kmeans(Data1.Age45,2,nstart = 25,iter.max = 15)
#K means for cluster N = 3
km.Data1.Age45 = kmeans(Data1.Age45,3,nstart = 25,iter.max = 15)
#K means for cluster N = 4
km.Data1.Age45 = kmeans(Data1.Age45,4,nstart = 25,iter.max = 15)
#K means for cluster N = 5
km.Data1.Age45 = kmeans(Data1.Age45,5,nstart = 25,iter.max = 15)
#K means for cluster N = 6
km.Data1.Age45 = kmeans(Data1.Age45,6,nstart = 25,iter.max = 15)
#K means for cluster N = 7
km.Data1.Age45 = kmeans(Data1.Age45,7,nstart = 25,iter.max = 15)
#K means for cluster N = 8
km.Data1.Age45 = kmeans(Data1.Age45,8,nstart = 25,iter.max = 15)
#K means for cluster N = 9
km.Data1.Age45 = kmeans(Data1.Age45,9,nstart = 25,iter.max = 15)
#K means for cluster N = 10
km.Data1.Age45 = kmeans(Data1.Age45,10,nstart = 25,iter.max = 15)

```

K-median

```

#K-median for 2-10 Clusters
kmed.Mart1.Age45.2 <- kGmedian(Data1.Age45, ncenters = 2)
kmed.Mart1.Age45.3 <- kGmedian(Data1.Age45, ncenters = 3)

```

```

kmed.Mart1.Age45.4 <- kGmedian(Data1.Age45, ncenters = 4)
kmed.Mart1.Age45.5 <- kGmedian(Data1.Age45, ncenters = 5)
kmed.Mart1.Age45.6 <- kGmedian(Data1.Age45, ncenters = 6)
kmed.Mart1.Age45.7 <- kGmedian(Data1.Age45, ncenters = 7)
kmed.Mart1.Age45.8 <- kGmedian(Data1.Age45, ncenters = 8)
kmed.Mart1.Age45.9 <- kGmedian(Data1.Age45, ncenters = 9)
kmed.Mart1.Age45.10 <- kGmedian(Data1.Age45, ncenters = 10)

```

KNN

```

set.seed(1234)
data_norm<-function(x){((x-min(x)/ (max(x)-min(x))))}
Data1.Age45_Norm<-as.data.frame(lapply(Data1.Age45, data_norm))
ind<-sample(2, nrow(Data1.Age45_Norm), replace=TRUE, prob = c(0.7,0.3))
train.data<-Data1.Age45_Norm[ind==1,]
test.data<-Data1.Age45_Norm[ind==2,]
predict.data<-knn(train.data, test.data, train.data[,6], k =50 , prob=TRUE)
table(predict.data,test.data[,6])

```

City Category A with Age 25:

Kmeans

```

#K Means#
#AGE 25#
#K means for cluster N = 2
km.cityA.age25 = kmeans(Data.A.Age25,2,nstart = 25,iter.max = 15)
#K means for cluster N = 3
km.cityA.age25 = kmeans(Data.A.Age25,3,nstart = 25,iter.max = 15)
#K means for cluster N = 4
km.cityA.age25 = kmeans(Data.A.Age25,4,nstart = 25,iter.max = 15)
#K means for cluster N = 5
km.cityA.age25 = kmeans(Data.A.Age25,5,nstart = 25,iter.max = 15)
#K means for cluster N = 6
km.cityA.age25 = kmeans(Data.A.Age25,6,nstart = 25,iter.max = 15)
#K means for cluster N = 7
km.cityA.age25 = kmeans(Data.A.Age25,7,nstart = 25,iter.max = 15)
#K means for cluster N = 8
km.cityA.age25 = kmeans(Data.A.Age25,8,nstart = 25,iter.max = 15)
#K means for cluster N = 9
km.cityA.age25 = kmeans(Data.A.Age25,9,nstart = 25,iter.max = 15)

```


K-median

#K-median for 2-10 Clusters

```
kmed.CityA.Age25.2 <- kGmedian(Data.A.Age25, ncenters = 2)
kmed.CityA.Age25.3 <- kGmedian(Data.A.Age25, ncenters = 3)
kmed.CityA.Age25.4 <- kGmedian(Data.A.Age25, ncenters = 4)
kmed.CityA.Age25.5 <- kGmedian(Data.A.Age25, ncenters = 5)
kmed.CityA.Age25.6 <- kGmedian(Data.A.Age25, ncenters = 6)
kmed.CityA.Age25.7 <- kGmedian(Data.A.Age25, ncenters = 7)
kmed.CityA.Age25.8 <- kGmedian(Data.A.Age25, ncenters = 8)
kmed.CityA.Age25.9 <- kGmedian(Data.A.Age25, ncenters = 9)
kmed.CityA.Age25.10 <- kGmedian(Data.A.Age25, ncenters = 10)
```

KNN

```
set.seed(1234)
data_norm<-function(x){((x-min(x)/ (max(x)-min(x))))}
Data.A.Age25_Norm<-as.data.frame(lapply(Data.A.Age25, data_norm))
ind<-sample(2, nrow(Data.A.Age25_Norm), replace=TRUE, prob = c(0.7,0.3))
train.data<-Data.A.Age25_Norm[ind==1,]
test.data<-Data.A.Age25_Norm[ind==2,]
predict.data<-knn(train.data, test.data, train.data[,5], k =50 , prob=TRUE)
table(predict.data,test.data[,5])
```

Occupation 0

For k = 2

Cluster Division	K-means	K-median
1	452	452
2	20225	20225

For k =3

Cluster Division	K-means	K-median
1	8155	8180
2	12070	12045

3	452	452
---	-----	-----

For k=4

Cluster Division	K-means	K-median
1	6105	6871
2	452	6023
3	8973	452
4	5147	7331

For k=5

Cluster Division	K-means	K-median
1	6589	452
2	2766	2766
3	5373	6589
4	5497	5497
5	452	5373

For k=6

Cluster Division	K-means	K-median
1	4996	452
2	4872	5055
3	2183	4930
4	3119	2161
5	5055	3083
6	452	4996

For k=7

Cluster Division	K-means	K-median
1	3002	3002
2	2123	2123
3	5101	3894
4	2985	5101
5	3894	3120
6	452	452
7	3120	2985

For k=8

Cluster Division	K-means	K-median
1	3038	3216
2	2985	3879
3	2618	2985
4	3228	2618
5	3879	2655
6	452	1834
7	2840	452
8	1637	3038

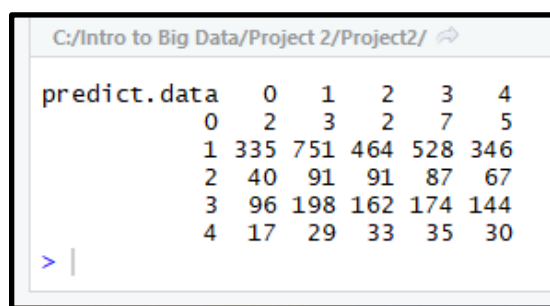
For k=9

Cluster Division	K-means	K-median
1	1362	2242
2	452	1404
3	3879	2985
4	2381	2371
5	2934	3038

6	1404	452
7	3038	3879
8	2242	2944
9	2985	1362

For k=10

Cluster Division	K-means	K-median
1	1232	452
2	2242	1486
3	3797	2944
4	452	2761
5	1362	2371
6	1404	1830
7	2932	2242
8	2381	1404
9	2934	1362
10	1941	3825



```

C:/Intro to Big Data/Project 2/Project2/
predict.data  0   1   2   3   4
              0   2   3   2   7   5
              1 335 751 464 528 346
              2  40  91  91  87  67
              3  96 198 162 174 144
              4  17  29  33  35  30
> |

```

KNN

Conclusion:

From the above table, we can conclude from the calculations made on the number of elements specified as belonging to a specific cluster, based on the two clustering methods, varying number of clusters, and varying dataset distribution, allow us to further explain our experiments in clustering, and support some points made earlier. From the above data, we can clearly understand that how $k=4$ is an optimal value for the clustering. It gives us similar result in K means as well as K median. The other cluster values are not that reliable. We can clearly understand how $k=4$ is a good clustering value in compared to others.

Clustering for Marital Status 1 and Age 45

For $k=2$

Cluster Division	K-means	K-median
1	12289	285
2	285	12289

For $k=3$

Cluster Division	K-means	K-median
1	7332	285
2	4957	7332
3	285	4957

For $k=4$

Cluster Division	K-means	K-median
1	5383	3276
2	285	3630
3	3630	5383
4	3276	285

For $k=5$

Cluster Division	K-means	K-median
1	3074	2931
2	1924	3074
3	2931	1924
4	4360	4360
5	285	285

For k=6

Cluster Division	K-means	K-median
1	285	1864
2	2822	3878
3	3533	2922
4	1424	2549
5	2646	1076
6	1864	285

For k=7

Cluster Division	K-means	K-median
1	1686	1692
2	1845	3485
3	1944	1845
4	3486	1076
5	285	285
6	1076	1938
7	2252	2253

For k=8

Cluster Division	K-means	K-median
1	2122	285
2	1021	1858
3	1944	1690
4	1384	1926
5	285	1815
6	3249	2620
7	883	1304
8	1686	1076

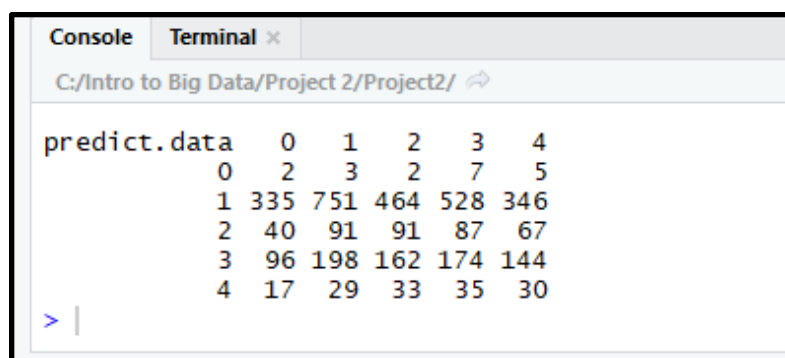
For k=9

Cluster Division	K-means	K-median
1	285	1655
2	2620	1132
3	1931	1021
4	1372	1690
5	883	883
6	1645	1362
7	1685	2620
8	1132	285
9	1021	1926

For k=10

Cluster Division	K-means	K-median
1	1645	883
2	2606	1096

3	285	703
4	831	285
5	883	1363
6	1021	1690
7	1589	1060
8	1210	1926
9	1372	948
10	1132	2620



```

Console Terminal x
C:/Intro to Big Data/Project 2/Project2/
predict.data  0  1  2  3  4
              0  2  3  2  7  5
              1 335 751 464 528 346
              2  40  91  91  87  67
              3  96 198 162 174 144
              4  17  29  33  35  30
> |

```

KNN

Conclusion:

From the above table, we can conclude from the calculations made on the number of elements specified as belonging to a specific cluster, based on the two clustering methods, varying number of clusters, and varying dataset distribution, allow us to further explain our experiments in clustering, and support some points made earlier. From the above data, we can clearly understand that how $k=4$ is an optimal value for the clustering. It gives us similar result in K means as well as K median. The other cluster values are not that reliable. We can clearly understand how $k=4$ is a good clustering value in compared to others.

Clustering for City Category A with Age 25

For $k = 2$

Cluster Division	K-means	K-median
------------------	---------	----------

1	7749	191
2	191	7749

For k =3

Cluster Division	K-means	K-median
1	191	4602
2	3147	3147
3	4602	191

For k=4

Cluster Division	K-means	K-median
1	2414	2414
2	191	191
3	3461	1874
4	1874	3461

For k=5

Cluster Division	K-means	K-median
1	191	2100
2	980	2448
3	2448	191
4	2221	2221
5	2100	980

For k=6

Cluster Division	K-means	K-median
1	2132	753

2	191	2108
3	1688	191
4	1131	1624
5	754	1132
6	2044	2132

For k=7

Cluster Division	K-means	K-median
1	1930	1240
2	1195	1531
3	1121	1121
4	1219	753
5	191	191
6	753	1930
7	1531	1174

For k=8

Cluster Division	K-means	K-median
1	1288	667
2	1213	1005
3	859	1213
4	1440	1174
5	1033	958
6	742	191
7	1174	1440
8	191	1292

For k=9

Cluster Division	K-means	K-median
1	915	1213
2	460	499
3	1213	915
4	499	1440
5	859	859
6	1440	460
7	1174	191
8	191	1174
9	1189	1189

For k=10

Cluster Division	K-means	K-median
1	1050	1189
2	191	499
3	460	859
4	1189	915
5	499	1413
6	859	460
7	594	1159
57	757	396
9	915	859
10	1426	191

predict.data	0	1	2	3	4
0	25	43	19	26	21
1	323	658	344	298	259
2	29	87	43	42	37
3	27	33	24	20	22
4	5	9	1	10	6

KNN

Conclusion:

From the above table, we can conclude from the calculations made on the number of elements specified as belonging to a specific cluster, based on the two clustering methods, varying number of clusters, and varying dataset distribution, allow us to further explain our experiments in clustering, and support some points made earlier. From the above data, we can clearly understand that how $k=4$ is an optimal value for the clustering. It gives us similar result in K means as well as K median. The other cluster values are not that reliable. We can clearly understand how $k=4$ is a good clustering value in compared to others.

Knn: We use Knn Algorithm to Classify Data. Firstly, We sampled dataset into Training which consisted of 70% of the original Data and Test set with the rest 30%. We used normalization technique for better classification of Data. The knn function then classified the Test data based on the number of years lived in the city. To view the classification, we used the table command where we could clearly see how the subsequent data was classified based on the training set.

KGmedian: The procedure is similar to the kmeans clustering technique performed recursively with the MacQueen algorithm. The advantage of the kGmedian algorithm compared to MacQueen strategy is that it deals with sum of norms instead of sum of squared norms, ensuring a more robust behaviour against outlying values.

Deduction from Clustering Methods and Plots

- All four of the clustering methods used have returned satisfactory results.
 - We found that K-Means is the best when trying to look at general trends in the data. It was easier to implement and was computationally faster.
 - However, the other hierarchical clustering methods were also efficient at helping in the exploration of data.

- Through the numerous plots and analyses we have deduced the best solution.
 - We found that 4 clusters appear to be the most optimal. This has been further elaborated under the conclusion of each clustering method.
 - Using KNN, we noticed that 70:30 dataset gives the best result.
- The most effective way to understand the data is by visually representing them using plots.
 - For this project, we found that IClust was the most helpful and easy to understand plotting method. The main advantage of IClust is that since it is better able to identify media groups, the plots ended up looking concise yet readable.
 - The other plotting methods, though not as visually appealing as IClust, also contributed immensely in the understanding of the datasets, clusters and how to process it to achieve optimal results.

Training Data and Test Data for 70% and 30% for Occupation 0

```
Data.Occ0 <- subset(Data.clean,Occupation == "0")
head(Data.Occ0)
Occ0.train <- sample(nrow(Data.Occ0),0.7*nrow(Data.Occ0))
Occ0.clean.train <- Data.Occ0[Occ0.train,]
Occ0.test <- Data.Occ0[-Occ0.train,]
Occ0.clean.train$User_ID<- NULL
Occ0.test$User_ID<- NULL
head(Occ0.clean.train)
```

Training Data and Test Data for 60% and 40% for Occupation 0

```
Occ01.train <- sample(nrow(Data.Occ0),0.6*nrow(Data.Occ0))
Occ01.clean.train <- Data.Occ0[Occ01.train,]
Occ0.test <- Data.Occ0[-Occ01.train,]
Occ01.clean.train$User_ID<- NULL
Occ0.test$User_ID<- NULL
```

Training Data and Test Data for 50% and 50% for Occupation 0

```
Occ0.train <- sample(nrow(Data.Occ0),0.5*nrow(Data.Occ0))
Occ0.clean.train <- Data.Occ0[Occ0.train,]
Occ0.test <- Data.Occ0[-Occ0.train,]
Occ0.clean.train$User_ID<- NULL
Occ0.test$User_ID<- NULL
```

Training Data and Test Data for 70% and 30% for Marital Status 1 with Age 45

```
Data.Mart1 <- subset(Data.clean, Marital_Status == "1")
Data1.Age45 <- subset(Data.Mart1, Age=="45")
Data45.train <- sample(nrow(Data1.Age45), 0.7*nrow(Data1.Age45))
Data45.clean.train <- Data1.Age45[Data17.train,]
Data45.test <- Data1.Age17[-Data17.train,]
Data45.clean.train$User_ID <- NULL
Data45.test$User_ID <- NULL
```

Training Data and Test Data for 60% and 40% for Marital Status 1 with Age 45

```
Data.Mart1 <- subset(Data.clean, Marital_Status == "1")
Data1.Age45 <- subset(Data.Mart1, Age=="45")
Data45.train <- sample(nrow(Data1.Age45), 0.6*nrow(Data1.Age45))
Data45.clean.train <- Data1.Age45[Data17.train,]
Data45.test <- Data1.Age17[-Data17.train,]
Data45.clean.train$User_ID <- NULL
Data45.test$User_ID <- NULL
```

Training Data and Test Data for 50% and 50% for Marital Status 1 with Age 45

```
Data.Mart1 <- subset(Data.clean, Marital_Status == "1")
Data1.Age45 <- subset(Data.Mart1, Age=="45")
Data45.train <- sample(nrow(Data1.Age45), 0.5*nrow(Data1.Age45))
Data45.clean.train <- Data1.Age45[Data17.train,]
Data45.test <- Data1.Age17[-Data17.train,]
Data45.clean.train$User_ID <- NULL
Data45.test$User_ID <- NULL
```

Training Data and Test Data for 70% and 30% for City Category A with Age 25

```
Data.CityA <- subset(Data.clean, City_Category == "1")
Data.A.Age25 <- subset(Data.CityA, Age=="25")
Data25.train <- sample(nrow(Data.A.Age25), 0.7*nrow(Data.A.Age25))
Data25.clean.train <- Data.A.Age25[Data25.train,]
Data25.test <- Data.A.Age25[-Data25.train,]
Data25.clean.train$User_ID <- NULL
Data25.test$User_ID <- NULL
```

Training Data and Test Data for 60% and 40% for City Category A with Age 25

```

Data.CityA <- subset(Data.clean,City_Category == "1")
Data.A.Age25 <- subset(Data.CityA,Age=="25")
Data25.train <- sample(nrow(Data.A.Age25),0.6*nrow(Data.A.Age25))
Data25.clean.train <- Data.A.Age25[Data25.train,]
Data25.test <- Data.A.Age25[-Data25.train,]
Data25.clean.train$User_ID<- NULL
Data25.test$User_ID<- NULL

```

Training Data and Test Data for 50% and 50% for City Category A with Age 25

```

Data.CityA <- subset(Data.clean,City_Category == "1")
Data.A.Age25 <- subset(Data.CityA,Age=="25")
Data25.train <- sample(nrow(Data.A.Age25),0.5*nrow(Data.A.Age25))
Data25.clean.train <- Data.A.Age25[Data25.train,]
Data25.test <- Data.A.Age25[-Data25.train,]
Data25.clean.train$User_ID<- NULL
Data25.test$User_ID<- NULL

```

LM & GLM

In this, we have used the Gender as the dependent variable and taken Age, Product_ID, Marital_Status and Stay_in_Current_City_Years as independent variables.

LM is the linear model and GLM is generalized linear model.

LM: $Y = |a| + |b|x + \epsilon$

GLM: $Y = e^a \cdot e^b + \epsilon$

In this, we have used lm and glm methods to calculate the linear models and the coefficients of the independent variables.

For lm and glm methods, we have used the training and test data sets of 70:30 and explored their coefficients and plotting.

1). LM and GLM for Occupation 0**lm results:**

```
Call:
lm(formula = data[, "Gender"] ~ data[, "Age"] + data[, "Product_ID"] +
    data[, "Marital_Status"] + data[, "Stay_In_Current_City_Years"],
    data = Occ0.clean.train)

Coefficients:
              (Intercept)              data[, "Age"]
              -5.445e-15              4.263e-02
              data[, "Product_ID"]              data[, "Marital_Status"]
              -2.960e-02              9.582e-03
data[, "Stay_In_Current_City_Years"]
              -3.243e-02
```

```
Call: glm(formula = data[, "Gender"] ~ data[, "Age"] + data[, "Product_ID"] +
    data[, "Marital_Status"] + data[, "Stay_In_Current_City_Years"],
    family = gaussian, data = Occ0.clean.train)

Coefficients:
              (Intercept)              data[, "Age"]
              -5.445e-15              4.263e-02
              data[, "Product_ID"]              data[, "Marital_Status"]
              -2.960e-02              9.582e-03
data[, "Stay_In_Current_City_Years"]
              -3.243e-02

Degrees of Freedom: 20676 Total (i.e. Null); 20672 Residual
Null Deviance: 20680
Residual Deviance: 20590 AIC: 58610
```

glm results:

```
Call: glm(formula = data[, "Gender"] ~ data[, "Age"] + data[, "Product_ID"] +
    data[, "Marital_Status"] + data[, "Stay_In_Current_City_Years"],
    family = gaussian, data = Occ0.clean.train)

Coefficients:
              (Intercept)              data[, "Age"]
              -5.445e-15              4.263e-02
              data[, "Product_ID"]              data[, "Marital_Status"]
              -2.960e-02              9.582e-03
data[, "Stay_In_Current_City_Years"]
              -3.243e-02

Degrees of Freedom: 20676 Total (i.e. Null); 20672 Residual
Null Deviance: 20680
Residual Deviance: 20590 AIC: 58610
```

Occupation 0	lm Coefficients
Age	4.263e-02
Product_ID	-2.960e-02
Marital_Status	9.582e-03
Stay_In_Current_City_Years	-3.243e-02

Occupation 0	glm Coefficients
Age	4.263e-02
Product_ID	-2.960e-02
Marital_Status	9.582e-03
Stay_In_Current_City_Years	-3.243e-02

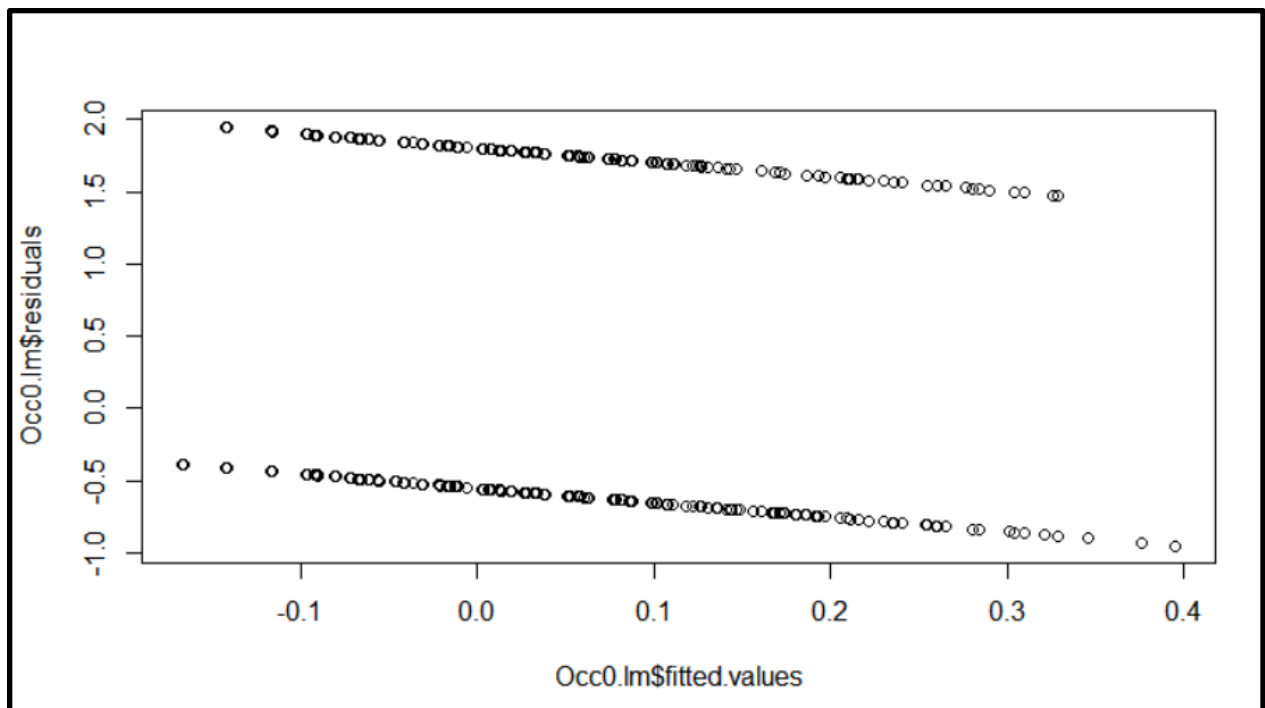
glm data:

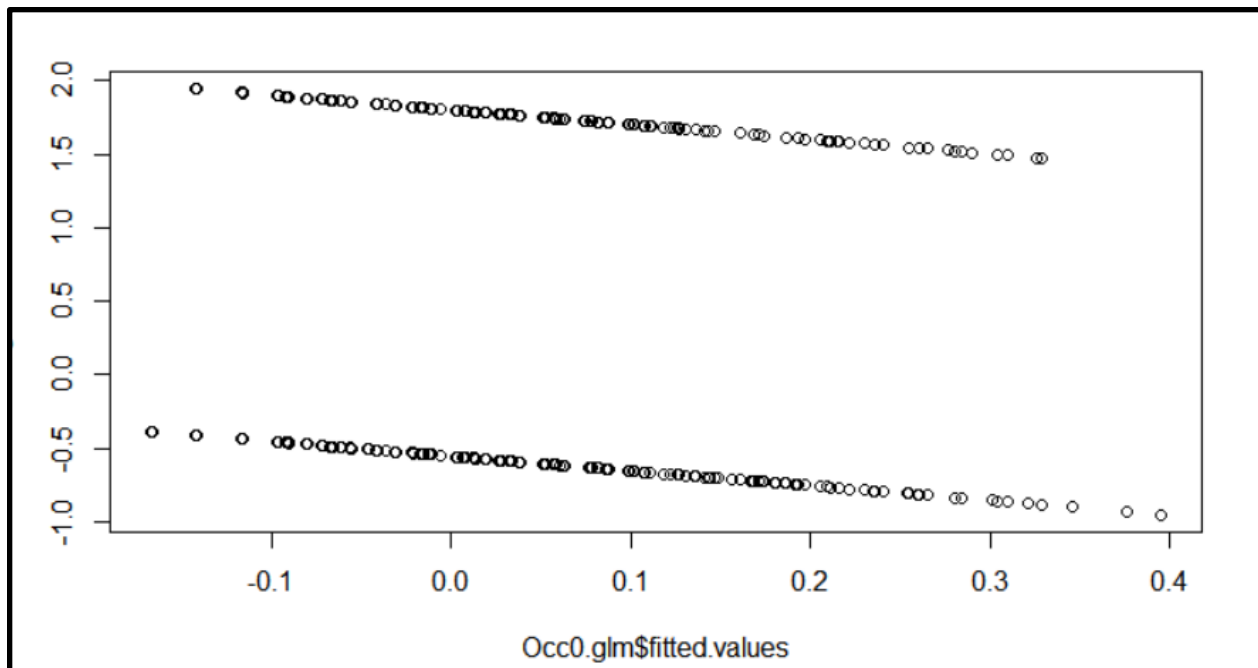
Degrees of Freedom: 20676 Total (i.e. Null); 20672 Residual

Null Deviance: 20680

Residual Deviance: 20590 AIC: 58610

The coefficients above, we can conclude that the equations tell us the change incurred to the dependent variable (Gender), by the Independent Variables listed above. The coefficients of GLM are in the scale of LM. And we can see how this affects the fitted values with the plots on this and the next page.





These plots are the residual values and fitted values plots of lm and glm for Occupation 0.

2). LM and GLM for Marital Status 1 with Age 45

lm results:

```
Call:
lm(formula = data[, "Gender"] ~ data[, "Product_ID"] + data[,
  "Stay_In_Current_City_Years"], data = Data45.clean.train)

Coefficients:
              (Intercept)              data[, "Product_ID"]
              -5.401e-15              -2.960e-02
data[, "Stay_In_Current_City_Years"]
              -3.073e-02
```

glm results:

```
Call: glm(formula = data[, "Gender"] ~ data[, "Product_ID"] + data[,
      "Stay_In_Current_City_Years"], family = gaussian, data = Data45.clean.train)

Coefficients:
              (Intercept)
              -5.401e-15
            data[, "Product_ID"]
              -2.960e-02
            data[, "Stay_In_Current_City_Years"]
              -3.073e-02

Degrees of Freedom: 20676 Total (i.e. Null); 20674 Residual
Null Deviance:      20680
Residual Deviance: 20640      AIC: 58650
```

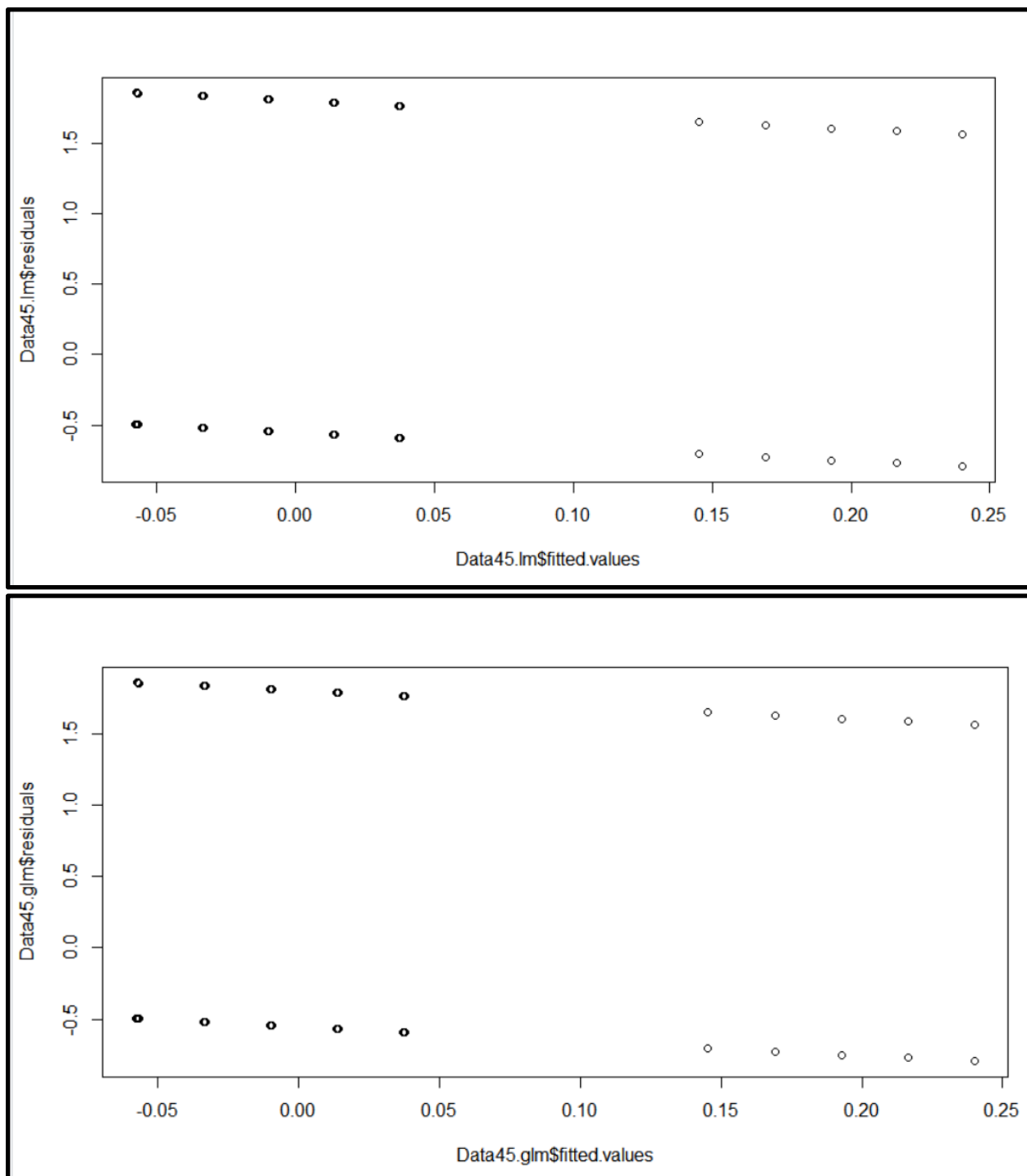
Martial Status 1 with Age 17	lm Coefficients
Product_ID	-2.960e-02
Stay_In_Current_City_Years	-3.073e-02

Martial Status 1 with Age 17	glm Coefficients
Product_ID	-2.960e-02
Stay_In_Current_City_Years	-3.073e-02

glm data:

Degrees of Freedom: 20676 Total (i.e. Null); 20674 Residual
 Null Deviance: 20680
 Residual Deviance: 20640 AIC: 58650

The the coefficients above, we can conclude that the equations tell us the change incurred to the dependent variable (Gender), by the Independent Variables listed above. The coefficients of GLM are in the scale of LM. And we can see how this affects the fitted values with the plots on this and the next page.



These plots are the residual values and fitted values plots of lm and glm for Marital Status 1 with Age 17.

3). LM and GLM for City Category A with Age 25

lm results:

```
Call:
lm(formula = data[, "Gender"] ~ data[, "Marital_Status"] + data[,
  "Product_ID"] + data[, "Stay_In_Current_City_Years"], data = Data25.clean.train)

Coefficients:
              (Intercept)              data[, "Marital_Status"]
              -5.522e-15              2.138e-02
data[, "Product_ID"] data[, "Stay_In_Current_City_Years"]
              -2.961e-02              -3.116e-02
```

glm results:

```
Call: glm(formula = data[, "Gender"] ~ data[, "Marital_Status"] + data[,
  "Product_ID"] + data[, "Stay_In_Current_City_Years"], family = gaussian,
  data = Data25.clean.train)

Coefficients:
              (Intercept)              data[, "Marital_Status"]
              -5.522e-15              2.138e-02
data[, "Product_ID"] data[, "Stay_In_Current_City_Years"]
              -2.961e-02              -3.116e-02

Degrees of Freedom: 20676 Total (i.e. Null); 20673 Residual
Null Deviance: 20680
Residual Deviance: 20630 AIC: 58640
```

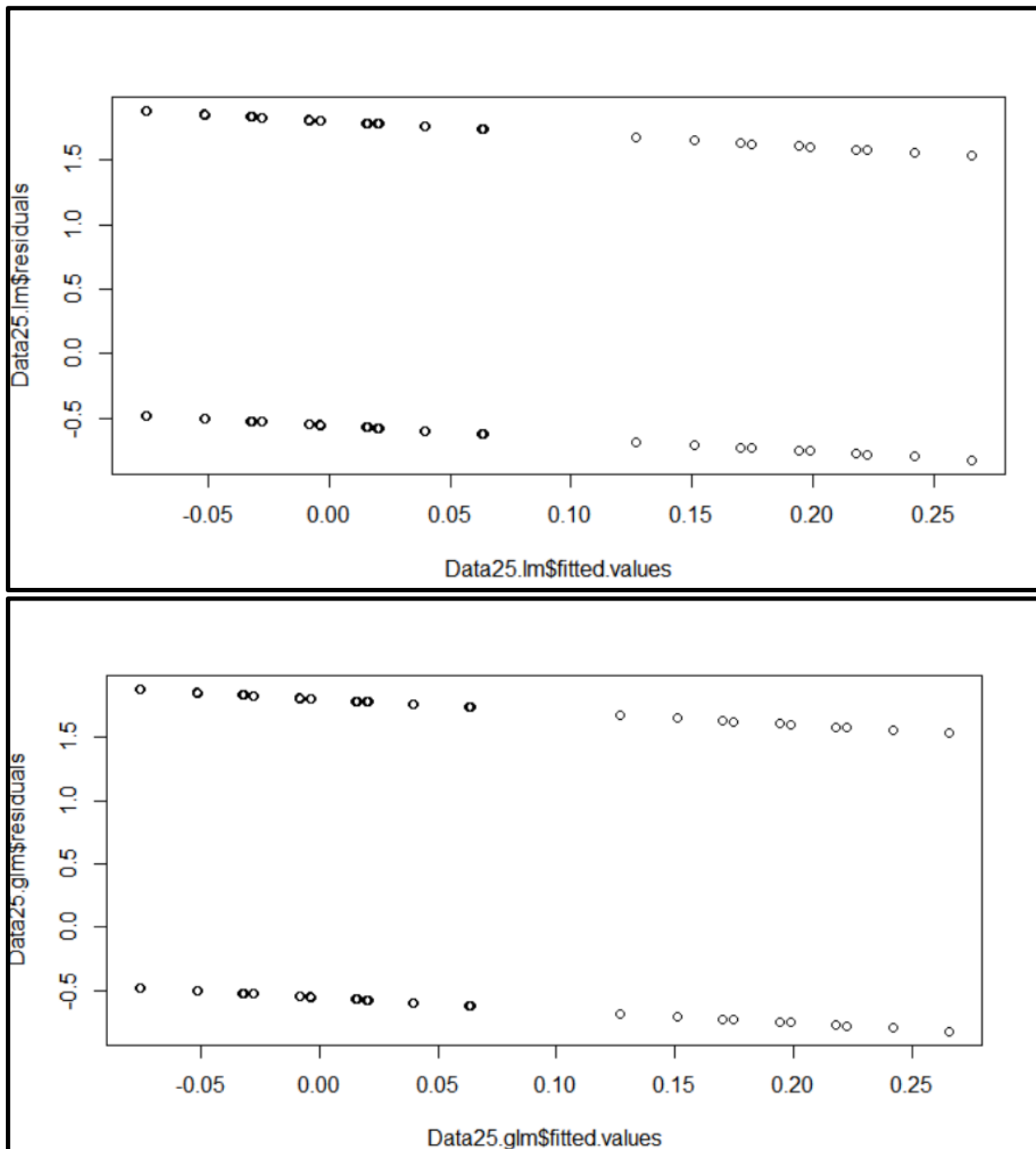
City A with Age 25	lm Coefficients
Marital_Status	2.138e-02
Product_ID	-2.961e-02
Stay_In_Current_City_Years	-3.116e-02

City A with Age 25	glm Coefficients
Marital_Status	2.138e-02
Product_ID	-2.961e-02
Stay_In_Current_City_Years	-3.116e-02

glm data:

Degrees of Freedom: 20676 Total (i.e. Null); 20673 Residual
 Null Deviance: 20680
 Residual Deviance: 20630 AIC: 58640

The the coefficients above, we can conclude that the equations tell us the change incurred to the dependent variable (Gender), by the Independent Variables listed above. The coefficients of GLM are in the scale of LM. And we can see how this affects the fitted values with the plots on this and the next page.



These plots are the residual values and fitted values plots of lm and glm for City Category A with Age 25.

Predict

Prediction lm in Occupation 0

```
> autopredict
```

	fit	lwr	upr
1	0.077446413	-1.879126	2.034019
2	0.002554984	-1.953963	1.959073
3	0.002783114	-1.953735	1.959301
4	0.003180488	-1.953337	1.959698
5	0.002532508	-1.953986	1.959051
6	0.002572965	-1.953945	1.959091
7	0.002404620	-1.954114	1.958923
8	0.003109914	-1.953408	1.959628
9	0.003110364	-1.953408	1.959628
10	-0.015990665	-1.972556	1.940575
11	-0.005816227	-1.962274	1.950642
12	-0.005576633	-1.962034	1.950881
13	-0.005993786	-1.962452	1.950464
14	-0.005882081	-1.962340	1.950576
15	-0.005559552	-1.962017	1.950898
16	-0.005560001	-1.962018	1.950898
17	-0.006034018	-1.962492	1.950424
18	-0.005850839	-1.962309	1.950607
19	-0.005814204	-1.962272	1.950644
20	-0.006070654	-1.962529	1.950387
21	-0.006071103	-1.962529	1.950387
22	-0.005949733	-1.962408	1.950508
23	-0.005767903	-1.962226	1.950690
24	-0.005850390	-1.962308	1.950607
25	-0.005856908	-1.962315	1.950601

Prediction glm in Occupation 0:

```
> PredictOccu0 <- predict.glm(Occ0.glm,NULL,type=c("link","response","terms"))
> PredictOccu0
```

1	2	3	4	5	6
0.077446413	0.002554984	0.002783114	0.003180488	0.002532508	0.002572965
7	8	9	10	11	12
0.002404620	0.003109914	0.003110364	-0.015990665	-0.005816227	-0.005576633
13	14	15	16	17	18
-0.005993786	-0.005882081	-0.005559552	-0.005560001	-0.006034018	-0.005850839
19	20	21	22	23	24
-0.005814204	-0.006070654	-0.006071103	-0.005949733	-0.005767903	-0.005850390
25	26	27	28	29	30
-0.005856908	0.062507066	0.062613153	0.007269210	0.013249152	-0.036226627
31	32	33	34	35	36
-0.036722220	0.081777650	0.013802509	0.013774639	0.013164867	0.013668328
37	38	39	40	41	42
-0.005758688	-0.060889109	-0.061308060	0.087474322	0.019025839	0.019154401
43	44	45	46	47	48
0.019361179	0.018765118	0.018974818	0.052666157	0.052694476	0.053000373
49	50	51	52	53	54
0.052928001	0.052807755	0.053019702	0.052279796	0.052864394	0.052423192
55	56	57	58	59	60
0.052981044	0.052831804	0.053055214	-0.041056590	-0.040814749	-0.041098395
61	62	63	64	65	66
0.052981123	0.052831804	0.053055214	-0.041056590	-0.040814749	-0.041098395
67	68	69	70	71	72
-0.091144837	-0.090611259	-0.090940756	0.171605954	-0.031031616	-0.030912944
73	74	75	76	77	78
0.013798463	-0.036206174	-0.036201903	-0.041069851	-0.090863888	-0.090705658
79	80	81	82	83	84
0.080527399	0.027454588	0.027622033	0.027648555	0.027833531	0.038233490
85	86	87	88	89	90
-0.041275280	-0.041358891	-0.040804860	-0.041280899	-0.041156608	-0.040750918
91	92	93	94	95	96
0.099014095	0.098874969	0.058406882	0.058411826	0.058288883	0.058373168
97	98	99	100	101	102
0.058304167	0.057811045	0.057992651	0.058457452	0.058094242	0.058538366
103	104	105	106	107	108
0.260612016	0.058131102	-0.080371416	-0.016576387	-0.016548292	-0.016072702
109	110	111	112	113	114
0.016349605	-0.015896041	-0.016447825	-0.016043708	-0.015802541	-0.015890422
115	116	117	118	119	120
-0.016149794	-0.016397479	-0.016113158	-0.016261499	0.018863338	0.018765118
121	122	123	124	125	126
0.221481804	-0.016541636	-0.080777556	-0.081125483	-0.080581341	-0.080762497
127	128	129	130	131	132
-0.080624270	-0.080884316	-0.005652152	-0.005774421	-0.021616763	0.033473564
133	134	135	136	137	138
139	140	141	142	143	144

Prediction lm for Marital Status 1 with Age 45:

```
> PredictData45
```

	fit	lwr	upr
1	0.037388664	-1.921107	1.995884
2	-0.033587201	-1.992035	1.924861
3	-0.033359068	-1.991807	1.925089
4	-0.032961690	-1.991410	1.925486
5	-0.033609678	-1.992058	1.924839
6	-0.033569220	-1.992017	1.924879
7	-0.033737567	-1.992186	1.924711
8	-0.033032265	-1.991480	1.925416
9	-0.033031815	-1.991480	1.925416
10	0.037744462	-1.920751	1.996240
11	0.014011144	-1.944412	1.972435
12	0.014250740	-1.944173	1.972674
13	0.013833583	-1.944590	1.972257
14	0.013945289	-1.944478	1.972369
15	0.014267822	-1.944156	1.972691
16	0.014267373	-1.944156	1.972691
17	0.013793350	-1.944630	1.972217
18	0.013976531	-1.944447	1.972400
19	0.014013167	-1.944410	1.972437
20	0.013756714	-1.944667	1.972180
21	0.013756265	-1.944667	1.972180
22	0.013877636	-1.944546	1.972301
23	0.014050468	-1.944364	1.972483

Prediction glm for Marital Status 1 with Age 45:


```
> PredictData45 <- predict.glm(Data45.glm,NULL,type=c("link","response","terms"))
> PredictData45
```



1	2	3	4	5	6
0.037388664	-0.033587201	-0.033359068	-0.032961690	-0.033609678	-0.033569220
7	8	9	10	11	12
-0.033737567	-0.033032265	-0.033031815	0.037744462	0.014011144	0.014250740
13	14	15	16	17	18
0.013833583	0.013945289	0.014267822	0.014267373	0.013793350	0.013976531
19	20	21	22	23	24
0.014013167	0.013756714	0.013756265	0.013877636	0.014059468	0.013976981
25	26	27	28	29	30
0.013970462	0.013864824	0.013970912	-0.056911902	0.013622082	-0.033243316
31	32	33	34	35	36
-0.033738916	0.013680969	0.014175445	0.014147575	0.013537796	0.014041262
37	38	39	40	41	42
-0.033737567	-0.033032265	-0.033031815	0.037744462	0.014011144	0.014250740
13	14	15	16	17	18
0.013833583	0.013945289	0.014267822	0.014267373	0.013793350	0.013976531
19	20	21	22	23	24
0.014013167	0.013756714	0.013756265	0.013877636	0.014059468	0.013976981
25	26	27	28	29	30
0.013970462	0.013864824	0.013970912	-0.056911902	0.013622082	-0.033243316
31	32	33	34	35	36
-0.033738916	0.013680969	0.014175445	0.014147575	0.013537796	0.014041262
37	38	39	40	41	42
0.014068683	-0.056600607	-0.057019563	0.037526893	0.037548020	0.037676584
43	44	45	46	47	48
0.037883365	0.037287297	0.037496999	0.013913598	0.013941918	0.014247818
49	50	51	52	53	54
0.014175445	0.014055198	0.014267148	0.013527232	0.014111838	0.013670630
55	56	57	58	59	60

Prediction lm for City Category A with Age 25:

```
> PredictData25
```

	fit	lwr	upr
1	0.063399498	-1.894765	2.021564
2	-0.008552684	-1.966665	1.949560
3	-0.008324468	-1.966437	1.949788
4	-0.007926945	-1.966039	1.950185
5	-0.008575168	-1.966688	1.949537
6	-0.008534696	-1.966647	1.949578
7	-0.008703104	-1.966816	1.949409
8	-0.007997545	-1.966110	1.950115
9	-0.007997096	-1.966109	1.950115
10	0.020357550	-1.937768	1.978483

156:1 (Top Level)  R Scri

Console ~/RProject2/  

```

8 -0.007997545 -1.966110 1.950115
9 -0.007997096 -1.966109 1.950115
10 0.020357550 -1.937768 1.978483
11 -0.003701234 -1.961757 1.954354
12 -0.003461551 -1.961517 1.954594
13 -0.003878860 -1.961934 1.954177
14 -0.003767113 -1.961822 1.954288
15 -0.003444463 -1.961500 1.954611
16 -0.003444912 -1.961500 1.954610
17 -0.003919107 -1.961975 1.954136
18 -0.003735860 -1.961791 1.954319
19 -0.003699210 -1.961755 1.954356
20 -0.003955756 -1.962011 1.954100
21 -0.003956206 -1.962012 1.954099
22 -0.003834791 -1.961890 1.954221
23 -0.003652892 -1.961708 1.954402

```

Prediction glm for City Category A with Age 25

```

> PredictData25 <- predict.glm(Data25.glm,NULL,type=c("link","response","terms"))
> PredictData25

```

1	2	3	4	5	6
0.063399498	-0.008552684	-0.008324468	-0.007926945	-0.008575168	-0.008534696
7	8	9	10	11	12
-0.008703104	-0.007997545	-0.007997096	0.020357550	-0.003701234	-0.003461551
13	14	15	16	17	18
-0.003878860	-0.003767113	-0.003444463	-0.003444912	-0.003919107	-0.003735860
19	20	21	22	23	24
-0.003699210	-0.003955756	-0.003956206	-0.003834791	-0.003652892	-0.003735410
25	26	27	28	29	30
-0.003741930	-0.003847607	-0.003741481	-0.032202702	0.039307438	-0.008208673
31	32	33	34	35	36
-0.008704453	0.039366347	0.039861002	0.039833122	0.039223122	0.039726771
37	38	39	40	41	42
-0.003643674	-0.031891294	-0.032310402	0.020139902	0.020161037	0.020289647
43	44	45	46	47	48
0.020496503	0.019900219	0.020109997	0.039599060	0.039627390	0.039933402
49	50	51	52	53	54
0.039861002	0.039740711	0.039952739	0.039212554	0.039797372	0.039356004
55	56	57	58	59	60
0.039914065	0.039764769	0.039988264	-0.003728665	-0.003486733	-0.003770485
61	62	63	64	65	66

Predict returns a vector of predicted values for the dependent variable based on training set.

What this project helped us in learning about Data Science?

This project has been an interesting and thorough learning approach towards data science. It gave us a glimpse into the world of data exploration and helped us understand its real-world applications. Though the project was challenging, it was overcoming these challenges that proved to be the driving factor in keeping our interest piqued throughout the duration of the project. Using and comparing different clustering methods and plots has been very helpful in understanding the underlying concepts of data science. This project has also given us an opportunity to work closely with R. It was riveting to be able to use our newly acquired knowledge of R and data science to work on various clustering methods. We found that data science involves a fair amount of experimentation before being able to come up with the optimal solution.