Assignment 7

Mohit Ghodake

April 18, 2019

##################  
#1  
#Creating Dummy Varaibles  
df$CHK0 <- factor(ifelse(df$CHK\_ACCT == 0, 1, 0))  
df$CHK1 <- factor(ifelse(df$CHK\_ACCT == 1, 1, 0))  
df$CHK2 <- factor(ifelse(df$CHK\_ACCT == 2, 1, 0))  
df$CHK3 <- factor(ifelse(df$CHK\_ACCT == 3, 1, 0))  
  
df$SAV0 <- factor(ifelse(df$SAV\_ACCT == 0, 1, 0))  
df$SAV1 <- factor(ifelse(df$SAV\_ACCT == 1, 1, 0))  
df$SAV2 <- factor(ifelse(df$SAV\_ACCT == 2, 1, 0))  
df$SAV3 <- factor(ifelse(df$SAV\_ACCT == 3, 1, 0))  
df$SAV4 <- factor(ifelse(df$SAV\_ACCT == 4, 1, 0))  
  
df$HISTORY0 <- factor(ifelse(df$HISTORY == 0, 1, 0))  
df$HISTORY1 <- factor(ifelse(df$HISTORY == 1, 1, 0))  
df$HISTORY2 <- factor(ifelse(df$HISTORY == 2, 1, 0))  
df$HISTORY3 <- factor(ifelse(df$HISTORY == 3, 1, 0))  
df$HISTORY4 <- factor(ifelse(df$HISTORY == 4, 1, 0))  
  
df$JOB0 <- factor(ifelse(df$JOB == 0, 1, 0))  
df$JOB1 <- factor(ifelse(df$JOB == 1, 1, 0))  
df$JOB2 <- factor(ifelse(df$JOB == 2, 1, 0))  
df$JOB3 <- factor(ifelse(df$JOB == 3, 1, 0))  
  
df$TYPE0 <- factor(ifelse(df$TYPE == 0, 1, 0))  
df$TYPE1 <- factor(ifelse(df$TYPE == 1, 1, 0))  
df$TYPE2 <- factor(ifelse(df$TYPE == 2, 1, 0))  
df$TYPE3 <- factor(ifelse(df$TYPE == 3, 1, 0))  
df$TYPE4 <- factor(ifelse(df$TYPE == 4, 1, 0))  
df$TYPE5 <- factor(ifelse(df$TYPE == 5, 1, 0))  
df$TYPE6 <- factor(ifelse(df$TYPE == 6, 1, 0))  
  
df$CHK\_ACCT <- NULL  
df$SAV\_ACCT <- NULL  
df$HISTORY <- NULL  
df$JOB <- NULL  
df$TYPE <- NULL

#########################  
#2  
#2  
set.seed(12345)  
df$PROFITABLE <- ifelse(df$NPV > 0, 1, 0)  
df$PROFITABLE <- factor(df$PROFITABLE)  
df1 <- df[, -16]  
df1 <- df1[, -41]  
  
kmeansCluster = kmeans(df1, 5, nstart=20)  
kmeansCluster

## K-means clustering with 5 clusters of sizes 287, 41, 124, 176, 372  
##   
## Cluster means:  
## AGE NUM\_CREDITS DURATION PRESENT\_RESIDENT EMPLOYMENT NUM\_DEPENDENTS  
## 1 34.46690 1.351916 19.96516 2.836237 2.365854 1.125436  
## 2 36.80488 1.390244 40.36585 2.975610 2.414634 1.121951  
## 3 36.98387 1.500000 33.12097 2.846774 2.403226 1.185484  
## 4 34.85795 1.426136 25.33523 2.920455 2.340909 1.193182  
## 5 36.08602 1.411290 13.31183 2.801075 2.408602 1.153226  
## RENT INSTALL\_RATE GUARANTOR OTHER\_INSTALL OWN\_RES TELEPHONE  
## 1 0.2090592 3.034843 0.04878049 0.1986063 0.7038328 0.4006969  
## 2 0.1219512 2.390244 0.00000000 0.2682927 0.5609756 0.8536585  
## 3 0.1774194 2.516129 0.03225806 0.2016129 0.6129032 0.5967742  
## 4 0.1647727 2.636364 0.05113636 0.1704545 0.7272727 0.4431818  
## 5 0.1693548 3.301075 0.06720430 0.1693548 0.7634409 0.2741935  
## FOREIGN REAL\_ESTATE AMOUNT\_REQUESTED CHK0 CHK1 CHK2  
## 1 0.02787456 0.28222997 2433.432 0.2648084 0.2543554 0.06968641  
## 2 0.02439024 0.04878049 12576.463 0.1951220 0.4878049 0.00000000  
## 3 0.01612903 0.10483871 7243.097 0.2741935 0.3387097 0.01612903  
## 4 0.02272727 0.18181818 4124.926 0.3125000 0.2443182 0.05681818  
## 5 0.05913978 0.41397849 1164.239 0.2715054 0.2446237 0.08333333  
## CHK3 SAV0 SAV1 SAV2 SAV3 SAV4 HISTORY0  
## 1 0.4111498 0.6062718 0.1010453 0.08013937 0.03484321 0.1777003 0.02787456  
## 2 0.3170732 0.5609756 0.1219512 0.04878049 0.02439024 0.2439024 0.12195122  
## 3 0.3709677 0.5564516 0.1048387 0.01612903 0.03225806 0.2903226 0.06451613  
## 4 0.3863636 0.6306818 0.1022727 0.04545455 0.04545455 0.1761364 0.06250000  
## 5 0.4005376 0.6075269 0.1021505 0.07526882 0.06720430 0.1478495 0.02150538  
## HISTORY1 HISTORY2 HISTORY3 HISTORY4 JOB0 JOB1  
## 1 0.02787456 0.5783972 0.08362369 0.2822300 0.003484321 0.15331010  
## 2 0.04878049 0.5121951 0.09756098 0.2195122 0.024390244 0.07317073  
## 3 0.06451613 0.3790323 0.18548387 0.3064516 0.008064516 0.09677419  
## 4 0.05113636 0.4943182 0.09090909 0.3011364 0.028409091 0.16477273  
## 5 0.05913978 0.5618280 0.05645161 0.3010753 0.037634409 0.30107527  
## JOB2 JOB3 TYPE0 TYPE1 TYPE2 TYPE3  
## 1 0.7108014 0.13240418 0.02439024 0.1916376 0.09059233 0.24041812  
## 2 0.3902439 0.51219512 0.14634146 0.2926829 0.19512195 0.07317073  
## 3 0.5806452 0.31451613 0.02419355 0.1854839 0.29838710 0.11290323  
## 4 0.6420455 0.16477273 0.03977273 0.1931818 0.14204545 0.26704545  
## 5 0.6048387 0.05645161 0.08602151 0.2956989 0.01881720 0.12903226  
## TYPE4 TYPE5 TYPE6  
## 1 0.3240418 0.03832753 0.09059233  
## 2 0.1219512 0.04878049 0.12195122  
## 3 0.1451613 0.08064516 0.15322581  
## 4 0.1931818 0.02840909 0.13636364  
## 5 0.3494624 0.05913978 0.06182796  
##   
## Clustering vector:  
## [1] 5 5 1 1 1 1 5 3 3 5 5 5 1 3 5 5 1 5 3 4 5 1 5 4 5 5 5 1 4 5 5 4 4 3  
## [35] 3 5 5 1 1 1 5 1 1 4 1 2 4 1 4 1 1 5 4 1 5 5 1 1 5 5 1 5 4 4 5 3 5 4  
## [69] 5 5 5 5 5 3 1 5 4 1 4 5 5 5 5 1 4 1 3 3 3 1 1 4 1 5 4 4 5 5 3 4 3 4  
## [103] 1 4 5 3 5 5 5 5 1 5 4 3 4 1 5 5 4 4 5 4 5 1 5 1 5 5 5 3 4 4 1 1 4 5  
## [137] 4 3 2 5 1 3 1 2 5 5 5 5 5 3 4 4 1 1 1 5 5 3 1 1 5 5 5 5 5 1 5 3 1 1  
## [171] 3 5 5 1 5 5 1 3 1 3 5 5 1 5 1 4 1 5 3 1 4 4 2 5 5 1 1 3 4 5 1 5 1 3  
## [205] 3 5 1 5 5 5 5 3 5 1 4 3 5 1 5 4 4 4 5 1 4 1 4 2 5 3 5 4 1 3 1 1 1 1  
## [239] 5 1 3 1 5 1 1 1 1 5 4 5 5 1 3 5 3 3 5 4 5 5 1 4 4 5 4 1 1 2 5 1 5 1  
## [273] 5 5 1 5 1 1 5 5 4 3 1 2 1 5 1 5 1 1 2 1 5 5 5 4 4 5 1 1 3 5 1 5 3 4  
## [307] 2 3 5 4 1 3 5 1 1 3 3 5 4 1 5 1 1 5 5 4 1 1 5 4 5 5 1 2 4 3 5 2 5 1  
## [341] 5 5 5 1 5 1 5 5 5 5 5 5 4 5 1 1 2 5 1 1 4 4 1 1 1 3 5 1 4 4 1 5 5 1  
## [375] 1 3 2 1 5 1 5 1 5 4 5 5 4 5 5 3 5 1 1 5 5 2 1 3 3 4 4 5 1 5 3 1 1 1  
## [409] 1 5 1 5 1 2 1 3 5 4 5 1 5 4 4 5 4 1 5 2 4 3 2 5 5 4 4 5 1 2 4 1 1 5  
## [443] 1 1 4 4 1 5 5 1 5 1 5 1 5 5 4 5 1 5 5 1 1 5 1 1 5 1 3 1 1 2 1 4 2 5  
## [477] 1 5 3 1 3 4 4 1 4 2 1 4 4 5 5 1 3 5 1 1 5 1 1 1 3 5 5 4 5 5 4 5 5 2  
## [511] 2 4 1 5 4 5 1 3 5 5 2 5 1 1 5 3 5 1 2 2 5 1 4 4 5 5 5 3 1 2 1 3 1 5  
## [545] 5 1 1 1 5 1 5 5 3 1 5 5 5 1 1 5 1 2 1 5 3 4 3 4 3 5 4 3 5 4 5 5 5 5  
## [579] 1 1 3 3 5 5 4 5 1 5 5 3 5 2 5 1 3 5 5 5 5 5 1 1 5 3 1 3 1 4 5 1 3 3  
## [613] 5 3 5 5 5 1 4 5 1 1 2 2 5 1 3 1 4 4 4 5 3 1 5 1 4 1 5 1 1 1 1 5 1 4  
## [647] 5 1 1 3 1 5 5 5 1 1 5 1 1 5 4 3 1 5 4 5 4 4 5 5 3 1 5 2 1 5 1 4 1 3  
## [681] 5 5 4 5 3 5 1 1 1 1 4 1 5 5 4 4 1 5 3 4 3 3 5 1 5 3 5 4 2 1 3 1 5 5  
## [715] 5 5 5 5 5 5 5 4 3 1 5 1 5 5 3 5 5 4 5 5 4 4 4 5 3 1 5 3 5 5 1 4 3 4  
## [749] 4 5 3 3 5 4 5 4 1 1 5 4 1 4 5 2 5 2 4 4 1 5 1 5 1 4 5 5 1 4 1 5 5 5  
## [783] 4 5 5 4 4 4 2 5 4 2 2 1 1 5 4 1 5 5 1 2 1 5 3 1 1 3 1 1 3 4 1 1 4 4  
## [817] 4 1 5 1 3 1 5 5 1 4 3 4 5 1 5 5 4 4 5 5 4 2 1 5 1 1 5 4 5 5 1 5 5 5  
## [851] 5 5 1 3 5 4 4 1 5 1 1 4 4 5 5 4 4 1 1 1 5 1 3 3 4 5 4 3 3 5 3 2 3 1  
## [885] 5 4 3 5 3 1 4 1 1 3 5 4 3 1 3 1 4 4 3 5 1 3 1 4 5 3 5 1 4 1 5 1 1 5  
## [919] 1 4 5 5 1 1 5 5 5 1 5 1 5 4 5 5 5 1 1 5 5 5 5 5 5 5 5 4 4 5 4 5 3 5  
## [953] 5 1 3 1 3 5 1 4 5 5 4 4 5 3 1 5 4 1 1 3 5 4 1 5 5 4 3 5 5 5 3 1 5 4  
## [987] 5 3 3 4 1 4 5 3 4 1 4 5 5 4  
##   
## Within cluster sum of squares by cluster:  
## [1] 50488779 164556980 145580619 66629463 45394979  
## (between\_SS / total\_SS = 94.1 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss"   
## [5] "tot.withinss" "betweenss" "size" "iter"   
## [9] "ifault"

dist(kmeansCluster$centers)

## 1 2 3 4  
## 2 10143.052   
## 3 4809.683 5333.372   
## 4 1691.503 8451.551 3118.181   
## 5 1269.211 11412.256 6078.890 2960.712

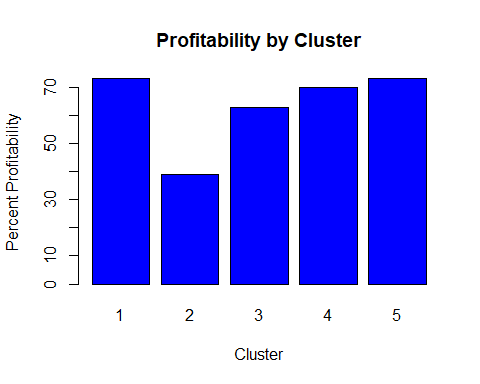
df$cluster <- kmeansCluster$cluster  
table(df$PROFITABLE, df$cluster)

##   
## 1 2 3 4 5  
## 0 77 25 46 53 100  
## 1 210 16 78 123 272

########################  
#4  
newdf <- data.frame(df$NPV, kmeansCluster$cluster)  
cluster1\_count = kmeansCluster$size[1]  
cluster2\_count = kmeansCluster$size[2]  
cluster3\_count = kmeansCluster$size[3]  
cluster4\_count = kmeansCluster$size[4]  
cluster5\_count = kmeansCluster$size[5]  
  
profitable\_cluster1\_count = 0  
profitable\_cluster2\_count = 0  
profitable\_cluster3\_count = 0  
profitable\_cluster4\_count = 0  
profitable\_cluster5\_count = 0  
  
NPV\_1 = 0  
NPV\_2 = 0  
NPV\_3 = 0  
NPV\_4 = 0  
NPV\_5 = 0  
  
  
for (i in 1:nrow(newdf)){  
 if (newdf[i, 2] == 1){   
 NPV\_1 = NPV\_1 + newdf[i, 1]  
 }  
 if (newdf[i, 2] == 2){   
 NPV\_2 = NPV\_2 + newdf[i, 1]  
 }  
 if (newdf[i, 2] == 3){  
 NPV\_3 = NPV\_3 + newdf[i, 1]  
 }  
 if (newdf[i, 2] == 4){   
 NPV\_4 = NPV\_4 + newdf[i, 1]  
 }  
 if (newdf[i, 2] == 5){   
 NPV\_5 = NPV\_5 + newdf[i, 1]  
 }  
}  
  
avgNPV\_1 <- NPV\_1/cluster1\_count  
avgNPV\_2 <- NPV\_2/cluster2\_count  
avgNPV\_3 <- NPV\_3/cluster3\_count  
avgNPV\_4 <- NPV\_4/cluster4\_count  
avgNPV\_5 <- NPV\_5/cluster5\_count  
  
for (i in 1:nrow(newdf)){  
 if (newdf[i, 2] == 1 & newdf[i, 1] > 0){   
 profitable\_cluster1\_count = profitable\_cluster1\_count + 1  
 }  
 if (newdf[i, 2] == 2 & newdf[i, 1] > 0){   
 profitable\_cluster2\_count = profitable\_cluster2\_count + 1  
 }  
 if (newdf[i, 2] == 3 & newdf[i, 1] > 0){   
 profitable\_cluster3\_count = profitable\_cluster3\_count + 1  
 }  
 if (newdf[i, 2] == 4 & newdf[i, 1] > 0){   
 profitable\_cluster4\_count = profitable\_cluster4\_count + 1  
 }  
 if (newdf[i, 2] == 5 & newdf[i, 1] > 0){   
 profitable\_cluster5\_count = profitable\_cluster5\_count + 1  
 }  
}  
  
percent\_prof\_1 <- profitable\_cluster1\_count/cluster1\_count\*100  
percent\_prof\_2 <- profitable\_cluster2\_count/cluster2\_count\*100  
percent\_prof\_3 <- profitable\_cluster3\_count/cluster3\_count\*100  
percent\_prof\_4 <- profitable\_cluster4\_count/cluster4\_count\*100  
percent\_prof\_5 <- profitable\_cluster5\_count/cluster5\_count\*100  
  
  
Cluster <- c(1,  
 2,  
 3,  
 4,  
 5)  
  
Count <- c(cluster1\_count,  
 cluster2\_count,  
 cluster3\_count,  
 cluster4\_count,  
 cluster5\_count)  
  
Profitable <- c(profitable\_cluster1\_count,  
 profitable\_cluster2\_count,  
 profitable\_cluster3\_count,  
 profitable\_cluster4\_count,  
 profitable\_cluster5\_count)  
  
PercentProfitable <- c(percent\_prof\_1,  
 percent\_prof\_2,  
 percent\_prof\_3,  
 percent\_prof\_4,  
 percent\_prof\_5)  
  
NPV <- c(NPV\_1,  
 NPV\_2,  
 NPV\_3,  
 NPV\_4,  
 NPV\_5)  
  
AVG\_NPV <- c(avgNPV\_1,  
 avgNPV\_2,  
 avgNPV\_3,  
 avgNPV\_4,  
 avgNPV\_5)  
  
df\_modified <- data.frame(Cluster, Count, Profitable, PercentProfitable, NPV, AVG\_NPV)  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 287 210 73.17073 20938 72.95470  
## 2 2 41 16 39.02439 -83273 -2031.04878  
## 3 3 124 78 62.90323 -40358 -325.46774  
## 4 4 176 123 69.88636 -6671 -37.90341  
## 5 5 372 272 73.11828 13483 36.24462

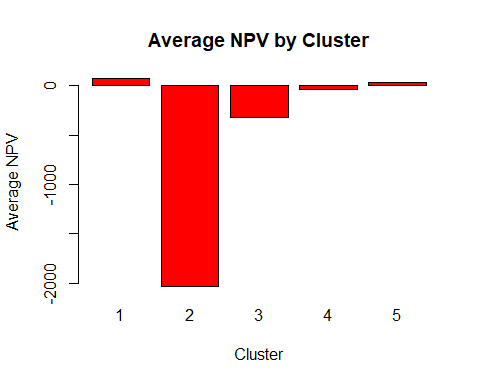
##  
#4a  
barplot(PercentProfitable, names.arg = Cluster, xlab = "Cluster", ylab = "Percent Profitability", col = "blue", main = "Profitability by Cluster")



##  
#4b  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 287 210 73.17073 20938 72.95470  
## 2 2 41 16 39.02439 -83273 -2031.04878  
## 3 3 124 78 62.90323 -40358 -325.46774  
## 4 4 176 123 69.88636 -6671 -37.90341  
## 5 5 372 272 73.11828 13483 36.24462

barplot(AVG\_NPV, names.arg = Cluster, xlab = "Cluster", ylab = "Average NPV", col = "red", main = "Average NPV by Cluster")



#######################  
#6  
kmeansCluster\_4 = kmeans(df1, 4, nstart=20)  
kmeansCluster\_4

## K-means clustering with 4 clusters of sizes 543, 285, 130, 42  
##   
## Cluster means:  
## AGE NUM\_CREDITS DURATION PRESENT\_RESIDENT EMPLOYMENT NUM\_DEPENDENTS  
## 1 35.61694 1.405157 15.06077 2.799263 2.375691 1.132597  
## 2 34.67368 1.375439 23.50526 2.908772 2.396491 1.189474  
## 3 36.80000 1.484615 33.34615 2.853846 2.376923 1.176923  
## 4 36.66667 1.404762 40.26190 2.976190 2.428571 1.142857  
## RENT INSTALL\_RATE GUARANTOR OTHER\_INSTALL OWN\_RES TELEPHONE  
## 1 0.1804788 3.261510 0.06261510 0.1749540 0.7495396 0.3167587  
## 2 0.1859649 2.712281 0.04561404 0.1859649 0.7157895 0.4245614  
## 3 0.1769231 2.538462 0.03846154 0.2076923 0.6000000 0.5769231  
## 4 0.1190476 2.357143 0.00000000 0.2619048 0.5714286 0.8571429  
## FOREIGN REAL\_ESTATE AMOUNT\_REQUESTED CHK0 CHK1 CHK2  
## 1 0.05340700 0.37384899 1469.365 0.2725599 0.2430939 0.07918969  
## 2 0.01754386 0.22456140 3583.589 0.2877193 0.2561404 0.06315789  
## 3 0.01538462 0.10000000 7127.523 0.2769231 0.3307692 0.01538462  
## 4 0.02380952 0.04761905 12511.714 0.1904762 0.5000000 0.00000000  
## CHK3 SAV0 SAV1 SAV2 SAV3 SAV4  
## 1 0.4051565 0.6206262 0.09760589 0.07366483 0.05340700 0.1546961  
## 2 0.3929825 0.6000000 0.10526316 0.06666667 0.04912281 0.1789474  
## 3 0.3769231 0.5538462 0.10769231 0.01538462 0.03076923 0.2923077  
## 4 0.3095238 0.5476190 0.14285714 0.04761905 0.02380952 0.2380952  
## HISTORY0 HISTORY1 HISTORY2 HISTORY3 HISTORY4 JOB0  
## 1 0.01841621 0.04972376 0.5524862 0.06813996 0.3112339 0.02762431  
## 2 0.05964912 0.04210526 0.5508772 0.08421053 0.2631579 0.01403509  
## 3 0.06153846 0.06153846 0.4000000 0.16923077 0.3076923 0.01538462  
## 4 0.11904762 0.04761905 0.5000000 0.11904762 0.2142857 0.02380952  
## JOB1 JOB2 JOB3 TYPE0 TYPE1 TYPE2  
## 1 0.25414365 0.6335175 0.08471455 0.06445672 0.2651934 0.02762431  
## 2 0.16491228 0.6771930 0.14385965 0.03157895 0.1929825 0.14385965  
## 3 0.08461538 0.5923077 0.30769231 0.03846154 0.1769231 0.30000000  
## 4 0.09523810 0.3809524 0.50000000 0.14285714 0.2857143 0.19047619  
## TYPE3 TYPE4 TYPE5 TYPE6  
## 1 0.16390424 0.3480663 0.05893186 0.0718232  
## 2 0.25614035 0.2350877 0.02105263 0.1192982  
## 3 0.12307692 0.1461538 0.07692308 0.1384615  
## 4 0.07142857 0.1190476 0.04761905 0.1428571  
##   
## Clustering vector:  
## [1] 1 1 1 1 1 1 1 3 3 1 1 1 1 3 1 1 2 1 3 2 1 2 1 2 1 1 1 2 2 1 1 3 2 3  
## [35] 3 1 1 2 1 1 1 1 2 2 1 4 2 2 2 1 2 1 3 1 1 1 2 2 1 1 1 1 2 2 1 3 1 2  
## [69] 1 1 1 1 1 3 1 1 2 2 2 1 1 1 1 1 2 1 3 3 3 1 2 2 1 1 2 2 1 1 3 2 3 2  
## [103] 2 2 1 3 1 1 1 1 1 1 2 3 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 3 2 3 1 1 2 1  
## [137] 2 3 4 1 1 3 1 4 1 1 1 1 1 3 3 2 2 2 1 1 1 3 2 1 1 1 1 1 1 1 1 3 1 2  
## [171] 3 1 1 1 1 1 1 3 1 3 1 1 2 1 2 2 1 1 3 1 2 2 4 1 1 1 1 3 2 1 1 1 1 3  
## [205] 3 1 1 1 1 1 1 3 1 2 2 3 1 1 1 2 2 2 1 1 2 1 2 4 1 3 1 2 2 3 2 2 1 1  
## [239] 1 2 3 1 1 1 2 2 2 1 2 1 1 2 3 1 3 3 1 3 1 1 1 2 2 1 2 1 1 4 1 2 1 1  
## [273] 1 1 2 1 1 2 1 1 2 3 2 4 1 1 1 1 1 1 4 2 1 1 1 2 2 1 2 1 3 1 1 1 3 2  
## [307] 4 3 1 2 1 3 1 1 1 3 3 1 2 1 1 1 2 1 1 2 2 2 1 2 1 1 2 4 2 3 1 4 1 1  
## [341] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 1 4 1 1 2 2 2 1 1 1 3 1 1 2 2 1 1 1 2  
## [375] 1 3 4 1 1 2 1 2 1 2 1 1 2 1 1 3 1 1 1 1 1 4 1 3 3 2 2 1 1 1 3 1 2 1  
## [409] 1 1 1 1 1 4 1 3 1 2 1 2 1 2 2 1 2 1 1 4 2 3 4 1 1 2 2 1 1 4 2 1 2 1  
## [443] 2 1 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 2 4 1 2 4 2 2 4 1  
## [477] 1 1 3 2 3 2 2 1 2 4 1 2 2 1 1 2 3 1 1 1 1 2 1 1 3 1 1 2 1 1 2 1 1 4  
## [511] 4 2 1 1 2 1 2 3 1 1 4 1 2 1 1 3 1 2 4 4 1 2 2 2 1 1 1 3 1 4 1 3 2 1  
## [545] 1 2 2 2 1 2 1 1 3 1 1 1 1 2 1 1 2 4 1 1 3 2 3 2 3 1 2 3 1 2 1 1 1 1  
## [579] 1 1 3 3 1 1 2 1 1 1 1 3 1 4 1 1 3 1 1 1 1 1 2 1 1 3 1 3 1 2 1 1 3 3  
## [613] 1 3 1 1 1 2 2 1 1 2 4 4 1 2 3 1 2 2 2 1 3 1 1 1 2 1 1 2 1 2 2 1 1 2  
## [647] 1 2 2 3 1 1 1 1 1 1 1 2 2 1 2 3 1 1 2 1 2 2 1 1 3 2 1 4 1 1 2 2 2 3  
## [681] 1 1 2 1 3 1 1 1 2 2 2 2 1 1 2 2 1 1 3 2 3 3 1 1 1 3 1 2 4 2 3 1 1 1  
## [715] 1 1 1 1 1 1 1 2 3 2 1 1 1 1 3 1 1 2 1 1 2 2 2 1 3 1 1 3 1 1 2 2 3 2  
## [749] 2 1 3 3 1 2 1 2 2 1 1 2 2 2 1 4 1 4 2 3 2 1 2 1 1 2 1 1 2 2 2 1 1 1  
## [783] 2 1 1 2 2 2 4 1 2 4 4 1 2 1 2 1 1 1 2 4 1 1 3 1 1 3 1 1 3 2 1 2 3 2  
## [817] 2 1 1 2 3 2 1 1 1 2 3 2 1 1 1 1 2 2 1 1 2 4 1 1 2 2 1 2 1 1 2 1 1 1  
## [851] 1 1 1 3 1 2 2 2 1 2 1 2 2 1 1 2 2 1 2 1 1 2 3 3 2 1 2 3 3 1 3 4 3 2  
## [885] 1 2 3 1 3 1 2 1 1 3 1 2 3 2 3 1 2 2 3 1 1 3 1 2 1 3 1 1 2 1 1 1 1 1  
## [919] 2 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 2 1 3 1  
## [953] 1 2 3 1 3 1 1 2 1 1 2 2 1 3 2 1 2 1 1 3 1 2 2 1 1 2 3 1 1 1 3 2 1 2  
## [987] 1 3 3 2 2 2 1 3 2 2 2 1 1 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 162713855 156315500 159056133 171776439  
## (between\_SS / total\_SS = 91.8 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss"   
## [5] "tot.withinss" "betweenss" "size" "iter"   
## [9] "ifault"

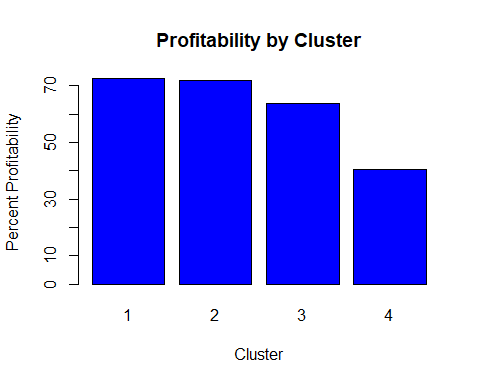
dist(kmeansCluster\_4$centers)

## 1 2 3  
## 2 2114.242   
## 3 5658.188 3543.948   
## 4 11042.379 8928.141 5384.196

newdf\_1 <- data.frame(df$NPV, kmeansCluster\_4$cluster)  
cluster1\_count = kmeansCluster\_4$size[1]  
cluster2\_count = kmeansCluster\_4$size[2]  
cluster3\_count = kmeansCluster\_4$size[3]  
cluster4\_count = kmeansCluster\_4$size[4]  
  
profitable\_cluster1\_count = 0  
profitable\_cluster2\_count = 0  
profitable\_cluster3\_count = 0  
profitable\_cluster4\_count = 0  
  
NPV\_1 = 0  
NPV\_2 = 0  
NPV\_3 = 0  
NPV\_4 = 0  
  
  
for (i in 1:nrow(newdf\_1)){  
 if (newdf\_1[i, 2] == 1){   
 NPV\_1 = NPV\_1 + newdf\_1[i, 1]  
 }  
 if (newdf\_1[i, 2] == 2){   
 NPV\_2 = NPV\_2 + newdf\_1[i, 1]  
 }  
 if (newdf\_1[i, 2] == 3){   
 NPV\_3 = NPV\_3 + newdf\_1[i, 1]  
 }  
 if (newdf\_1[i, 2] == 4){   
 NPV\_4 = NPV\_4 + newdf\_1[i, 1]  
 }  
}  
  
avgNPV\_1 <- NPV\_1/cluster1\_count  
avgNPV\_2 <- NPV\_2/cluster2\_count  
avgNPV\_3 <- NPV\_3/cluster3\_count  
avgNPV\_4 <- NPV\_4/cluster4\_count  
  
for (i in 1:nrow(newdf\_1)){  
 if (newdf\_1[i, 2] == 1 & newdf\_1[i, 1] > 0){   
 profitable\_cluster1\_count = profitable\_cluster1\_count + 1  
 }  
 if (newdf\_1[i, 2] == 2 & newdf\_1[i, 1] > 0){   
 profitable\_cluster2\_count = profitable\_cluster2\_count + 1  
 }  
 if (newdf\_1[i, 2] == 3 & newdf\_1[i, 1] > 0){   
 profitable\_cluster3\_count = profitable\_cluster3\_count + 1  
 }  
 if (newdf\_1[i, 2] == 4 & newdf\_1[i, 1] > 0){   
 profitable\_cluster4\_count = profitable\_cluster4\_count + 1  
 }  
}  
  
percent\_prof\_1 <- profitable\_cluster1\_count/cluster1\_count\*100  
percent\_prof\_2 <- profitable\_cluster2\_count/cluster2\_count\*100  
percent\_prof\_3 <- profitable\_cluster3\_count/cluster3\_count\*100  
percent\_prof\_4 <- profitable\_cluster4\_count/cluster4\_count\*100  
  
  
Cluster <- c(1,  
 2,  
 3,  
 4)  
  
Count <- c(cluster1\_count,  
 cluster2\_count,  
 cluster3\_count,  
 cluster4\_count)  
  
Profitable <- c(profitable\_cluster1\_count,  
 profitable\_cluster2\_count,  
 profitable\_cluster3\_count,  
 profitable\_cluster4\_count)  
  
PercentProfitable <- c(percent\_prof\_1,  
 percent\_prof\_2,  
 percent\_prof\_3,  
 percent\_prof\_4)  
  
NPV <- c(NPV\_1,  
 NPV\_2,  
 NPV\_3,  
 NPV\_4)  
  
AVG\_NPV <- c(avgNPV\_1,  
 avgNPV\_2,  
 avgNPV\_3,  
 avgNPV\_4)  
  
df\_modified <- data.frame(Cluster, Count, Profitable, PercentProfitable, NPV, AVG\_NPV)  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 543 394 72.55985 20548 37.841621  
## 2 2 285 205 71.92982 2703 9.484211  
## 3 3 130 83 63.84615 -37171 -285.930769  
## 4 4 42 17 40.47619 -81961 -1951.452381

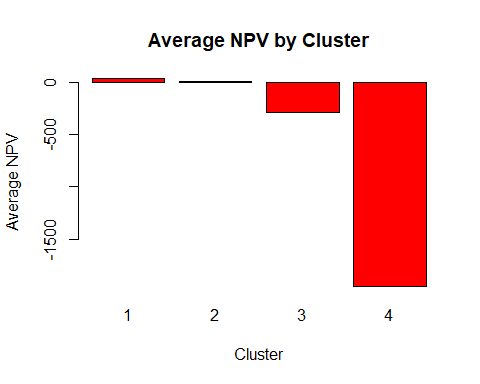
####  
barplot(PercentProfitable, names.arg = Cluster, xlab = "Cluster", ylab = "Percent Profitability", col = "blue", main = "Profitability by Cluster")



#####  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 543 394 72.55985 20548 37.841621  
## 2 2 285 205 71.92982 2703 9.484211  
## 3 3 130 83 63.84615 -37171 -285.930769  
## 4 4 42 17 40.47619 -81961 -1951.452381

barplot(AVG\_NPV, names.arg = Cluster, xlab = "Cluster", ylab = "Average NPV", col = "red", main = "Average NPV by Cluster")



#####################  
  
kmeansCluster\_6 = kmeans(df1, 6, nstart = 20)  
kmeansCluster\_6

## K-means clustering with 6 clusters of sizes 43, 370, 175, 283, 102, 27  
##   
## Cluster means:  
## AGE NUM\_CREDITS DURATION PRESENT\_RESIDENT EMPLOYMENT NUM\_DEPENDENTS  
## 1 33.79070 1.581395 38.18605 2.558140 2.395349 1.186047  
## 2 36.07568 1.408108 13.30000 2.802703 2.408108 1.154054  
## 3 34.86286 1.422857 24.77714 2.942857 2.354286 1.194286  
## 4 34.53357 1.360424 19.85866 2.816254 2.367491 1.123675  
## 5 37.67647 1.470588 32.68627 2.970588 2.401961 1.186275  
## 6 38.07407 1.259259 38.88889 3.074074 2.333333 1.074074  
## RENT INSTALL\_RATE GUARANTOR OTHER\_INSTALL OWN\_RES TELEPHONE  
## 1 0.06976744 2.139535 0.00000000 0.2558140 0.6744186 0.6744186  
## 2 0.16756757 3.302703 0.06486486 0.1648649 0.7648649 0.2729730  
## 3 0.16571429 2.617143 0.04571429 0.1714286 0.7371429 0.4685714  
## 4 0.21201413 3.053004 0.05300353 0.2014134 0.7031802 0.3886926  
## 5 0.19607843 2.676471 0.04901961 0.2058824 0.5882353 0.5784314  
## 6 0.18518519 2.370370 0.00000000 0.2222222 0.4814815 0.8518519  
## FOREIGN REAL\_ESTATE AMOUNT\_REQUESTED CHK0 CHK1 CHK2  
## 1 0.023255814 0.04651163 9400.860 0.2558140 0.3023256 0.00000000  
## 2 0.059459459 0.41351351 1160.824 0.2702703 0.2432432 0.08378378  
## 3 0.022857143 0.18857143 4040.926 0.3142857 0.2400000 0.05714286  
## 4 0.028268551 0.28621908 2411.735 0.2650177 0.2579505 0.07067138  
## 5 0.009803922 0.12745098 6669.882 0.2745098 0.3333333 0.01960784  
## 6 0.037037037 0.00000000 13611.296 0.1851852 0.6296296 0.00000000  
## CHK3 SAV0 SAV1 SAV2 SAV3 SAV4  
## 1 0.4418605 0.4883721 0.11627907 0.046511628 0.04651163 0.3023256  
## 2 0.4027027 0.6054054 0.10270270 0.075675676 0.06756757 0.1486486  
## 3 0.3885714 0.6400000 0.09142857 0.051428571 0.04571429 0.1714286  
## 4 0.4063604 0.6077739 0.10247350 0.077738516 0.03533569 0.1766784  
## 5 0.3725490 0.5784314 0.10784314 0.009803922 0.02941176 0.2745098  
## 6 0.1851852 0.5555556 0.14814815 0.037037037 0.00000000 0.2592593  
## HISTORY0 HISTORY1 HISTORY2 HISTORY3 HISTORY4 JOB0  
## 1 0.09302326 0.02325581 0.4651163 0.20930233 0.2093023 0.000000000  
## 2 0.02162162 0.05945946 0.5648649 0.05675676 0.2972973 0.037837838  
## 3 0.06857143 0.05142857 0.4857143 0.09142857 0.3028571 0.022857143  
## 4 0.02473498 0.02826855 0.5759717 0.08480565 0.2862191 0.003533569  
## 5 0.03921569 0.06862745 0.4019608 0.14705882 0.3431373 0.019607843  
## 6 0.18518519 0.07407407 0.4444444 0.11111111 0.1851852 0.037037037  
## JOB1 JOB2 JOB3 TYPE0 TYPE1 TYPE2  
## 1 0.09302326 0.5581395 0.34883721 0.02325581 0.2093023 0.37209302  
## 2 0.29729730 0.6081081 0.05675676 0.08648649 0.2972973 0.01891892  
## 3 0.17142857 0.6228571 0.18285714 0.02857143 0.2000000 0.14285714  
## 4 0.15901060 0.7173145 0.12014134 0.02473498 0.1908127 0.08480565  
## 5 0.08823529 0.5980392 0.29411765 0.04901961 0.1764706 0.26470588  
## 6 0.07407407 0.2962963 0.59259259 0.18518519 0.2962963 0.14814815  
## TYPE3 TYPE4 TYPE5 TYPE6  
## 1 0.09302326 0.11627907 0.06976744 0.11627907  
## 2 0.12972973 0.34864865 0.05945946 0.05945946  
## 3 0.26857143 0.19428571 0.02857143 0.13714286  
## 4 0.23674912 0.32862191 0.03886926 0.09540636  
## 5 0.13725490 0.16666667 0.06862745 0.13725490  
## 6 0.03703704 0.07407407 0.07407407 0.18518519  
##   
## Clustering vector:  
## [1] 2 2 4 4 4 4 2 1 5 2 2 2 4 1 2 2 4 2 1 3 2 4 2 3 2 2 2 4 3 2 2 5 3 5  
## [35] 5 2 2 4 4 4 2 4 4 3 4 1 3 4 3 4 4 2 5 4 2 2 4 3 2 2 4 2 3 3 2 5 2 3  
## [69] 2 2 2 2 2 5 4 2 3 4 3 2 2 2 2 4 3 4 5 5 5 4 4 3 4 2 3 3 2 2 1 3 5 3  
## [103] 4 3 2 5 2 2 2 2 4 2 3 5 3 4 2 2 3 3 2 3 2 4 2 4 2 2 2 1 3 5 4 4 3 2  
## [137] 3 5 6 2 4 5 4 1 2 2 2 2 2 1 5 3 4 4 4 2 2 5 4 4 2 2 2 2 2 4 2 5 4 4  
## [171] 5 2 2 4 2 2 4 1 4 5 4 2 4 2 4 3 4 2 1 4 3 3 6 2 2 4 4 5 3 2 4 2 4 5  
## [205] 5 2 4 2 2 2 2 5 2 4 3 5 2 4 2 3 3 3 2 4 3 4 3 6 2 5 2 3 4 5 4 3 4 4  
## [239] 2 3 1 4 2 4 4 4 4 2 3 2 2 4 5 2 5 5 2 5 2 2 4 3 3 2 3 4 4 1 2 4 2 4  
## [273] 2 2 4 2 4 3 2 2 3 1 4 6 4 2 4 2 4 4 1 4 2 2 2 3 3 2 4 4 5 2 4 2 5 3  
## [307] 6 5 2 3 4 5 2 4 4 5 5 2 3 4 2 4 4 2 2 3 4 4 2 3 2 2 4 6 3 5 2 6 2 4  
## [341] 2 2 2 4 2 4 2 2 2 2 2 2 3 2 4 4 6 2 4 4 3 3 4 4 4 1 2 4 3 3 4 2 2 4  
## [375] 4 5 6 4 2 4 2 4 2 3 2 2 3 2 2 5 2 4 4 2 2 1 4 5 1 3 3 2 4 2 5 4 4 4  
## [409] 4 2 4 2 4 1 4 5 2 3 2 4 2 3 3 2 3 4 2 6 3 5 1 2 2 3 3 2 4 6 3 4 4 2  
## [443] 4 4 3 3 4 2 2 4 2 4 2 4 2 2 3 2 4 2 2 4 4 2 4 4 2 4 1 4 4 6 4 3 6 2  
## [477] 4 2 5 4 5 3 3 4 3 1 4 3 3 2 2 4 5 2 4 4 2 4 4 4 1 2 2 3 2 2 3 2 2 6  
## [511] 6 3 4 2 3 2 4 5 2 2 1 2 4 4 2 5 2 4 1 1 2 4 3 3 2 2 2 5 4 6 4 5 4 2  
## [545] 2 4 4 4 2 4 2 2 5 4 2 2 2 4 4 2 4 6 4 2 5 3 5 3 5 2 3 5 2 3 4 2 2 2  
## [579] 4 4 1 5 2 2 3 2 4 2 2 1 2 6 2 4 1 2 2 2 2 2 4 4 2 5 4 5 4 3 2 4 5 5  
## [613] 2 1 2 2 2 4 3 2 4 4 1 6 2 4 5 4 3 3 3 2 5 4 2 4 3 4 2 4 4 4 4 2 4 3  
## [647] 2 4 4 1 4 2 2 2 4 4 2 4 4 2 3 1 4 2 3 2 3 3 2 2 5 4 2 6 4 2 4 3 4 5  
## [681] 2 2 3 2 5 2 4 4 4 4 3 4 2 2 3 3 4 2 5 3 5 1 2 4 2 1 2 3 6 4 5 4 2 2  
## [715] 2 2 2 2 2 2 2 3 5 4 2 4 2 2 5 2 2 3 2 2 3 3 3 2 5 4 2 1 2 2 4 3 5 3  
## [749] 3 2 5 5 2 3 2 3 4 4 2 3 4 3 2 6 2 1 3 5 4 2 4 2 4 3 2 2 4 3 4 2 2 2  
## [783] 3 2 2 3 3 3 6 2 3 6 1 4 4 2 3 4 2 2 4 6 4 2 5 4 4 1 4 4 5 3 4 4 5 3  
## [817] 3 4 2 4 5 4 2 2 4 3 1 3 2 4 2 2 3 3 2 2 3 6 4 2 3 4 2 3 2 2 4 2 2 2  
## [851] 2 2 4 5 2 3 3 4 2 4 4 3 3 2 2 3 3 4 4 4 2 4 5 5 3 2 3 5 5 2 1 6 1 4  
## [885] 2 3 5 2 5 4 3 4 4 5 2 3 5 4 5 4 3 3 5 2 4 5 4 3 2 5 2 4 3 4 2 4 4 2  
## [919] 3 3 2 2 4 4 2 2 2 4 2 4 2 3 2 2 2 4 4 2 2 2 2 2 2 2 2 3 3 2 3 2 1 2  
## [953] 2 4 1 4 5 2 4 3 2 2 3 3 2 5 4 2 3 4 4 5 2 3 4 2 2 3 5 2 2 2 5 4 2 3  
## [987] 2 5 5 3 4 3 2 5 3 4 3 2 2 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 39326283 44592167 57340162 47221131 52774509 77640416  
## (between\_SS / total\_SS = 96.0 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss"   
## [5] "tot.withinss" "betweenss" "size" "iter"   
## [9] "ifault"

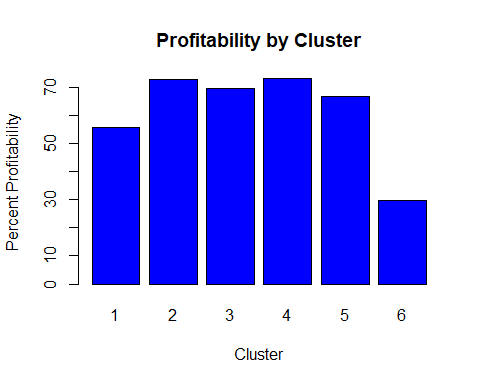
dist(kmeansCluster\_6$centers)

## 1 2 3 4 5  
## 2 8240.074   
## 3 5359.952 2880.125   
## 4 6989.150 1250.929 1629.198   
## 5 2730.987 5509.092 2628.970 4258.168   
## 6 4210.438 12450.499 9570.382 11199.578 6941.417

newdf\_2 <- data.frame(df$NPV, kmeansCluster\_6$cluster)  
cluster1\_count = kmeansCluster\_6$size[1]  
cluster2\_count = kmeansCluster\_6$size[2]  
cluster3\_count = kmeansCluster\_6$size[3]  
cluster4\_count = kmeansCluster\_6$size[4]  
cluster5\_count = kmeansCluster\_6$size[5]  
cluster6\_count = kmeansCluster\_6$size[6]  
  
profitable\_cluster1\_count = 0  
profitable\_cluster2\_count = 0  
profitable\_cluster3\_count = 0  
profitable\_cluster4\_count = 0  
profitable\_cluster5\_count = 0  
profitable\_cluster6\_count = 0  
  
NPV\_1 = 0  
NPV\_2 = 0  
NPV\_3 = 0  
NPV\_4 = 0  
NPV\_5 = 0  
NPV\_6 = 0  
  
  
for (i in 1:nrow(newdf\_2)){  
 if (newdf\_2[i, 2] == 1){   
 NPV\_1 = NPV\_1 + newdf\_2[i, 1]  
 }  
 if (newdf\_2[i, 2] == 2){   
 NPV\_2 = NPV\_2 + newdf\_2[i, 1]  
 }  
 if (newdf\_2[i, 2] == 3){   
 NPV\_3 = NPV\_3 + newdf\_2[i, 1]  
 }  
 if (newdf\_2[i, 2] == 4){   
 NPV\_4 = NPV\_4 + newdf\_2[i, 1]  
 }  
 if (newdf\_2[i, 2] == 5){   
 NPV\_5 = NPV\_5 + newdf\_2[i, 1]  
 }  
 if (newdf\_2[i, 2] == 6){   
 NPV\_6 = NPV\_6 + newdf\_2[i, 1]  
 }  
}  
  
avgNPV\_1 <- NPV\_1/cluster1\_count  
avgNPV\_2 <- NPV\_2/cluster2\_count  
avgNPV\_3 <- NPV\_3/cluster3\_count  
avgNPV\_4 <- NPV\_4/cluster4\_count  
avgNPV\_5 <- NPV\_5/cluster5\_count  
avgNPV\_6 <- NPV\_6/cluster6\_count  
  
for (i in 1:nrow(newdf\_2)){  
 if (newdf\_2[i, 2] == 1 & newdf\_2[i, 1] > 0){   
 profitable\_cluster1\_count = profitable\_cluster1\_count + 1  
 }  
 if (newdf\_2[i, 2] == 2 & newdf\_2[i, 1] > 0){   
 profitable\_cluster2\_count = profitable\_cluster2\_count + 1  
 }  
 if (newdf\_2[i, 2] == 3 & newdf\_2[i, 1] > 0){   
 profitable\_cluster3\_count = profitable\_cluster3\_count + 1  
 }  
 if (newdf\_2[i, 2] == 4 & newdf\_2[i, 1] > 0){   
 profitable\_cluster4\_count = profitable\_cluster4\_count + 1  
 }  
 if (newdf\_2[i, 2] == 5 & newdf\_2[i, 1] > 0){   
 profitable\_cluster5\_count = profitable\_cluster5\_count + 1  
 }  
 if (newdf\_2[i, 2] == 6 & newdf\_2[i, 1] > 0){   
 profitable\_cluster6\_count = profitable\_cluster6\_count + 1  
 }  
}  
  
percent\_prof\_1 <- profitable\_cluster1\_count/cluster1\_count\*100  
percent\_prof\_2 <- profitable\_cluster2\_count/cluster2\_count\*100  
percent\_prof\_3 <- profitable\_cluster3\_count/cluster3\_count\*100  
percent\_prof\_4 <- profitable\_cluster4\_count/cluster4\_count\*100  
percent\_prof\_5 <- profitable\_cluster5\_count/cluster5\_count\*100  
percent\_prof\_6 <- profitable\_cluster6\_count/cluster6\_count\*100  
  
  
Cluster <- c(1,  
 2,  
 3,  
 4,  
 5,  
 6)  
  
Count <- c(cluster1\_count,  
 cluster2\_count,  
 cluster3\_count,  
 cluster4\_count,  
 cluster5\_count,  
 cluster6\_count)  
  
Profitable <- c(profitable\_cluster1\_count,  
 profitable\_cluster2\_count,  
 profitable\_cluster3\_count,  
 profitable\_cluster4\_count,  
 profitable\_cluster5\_count,  
 profitable\_cluster6\_count)  
  
PercentProfitable <- c(percent\_prof\_1,  
 percent\_prof\_2,  
 percent\_prof\_3,  
 percent\_prof\_4,  
 percent\_prof\_5,  
 percent\_prof\_6)  
  
NPV <- c(NPV\_1,  
 NPV\_2,  
 NPV\_3,  
 NPV\_4,  
 NPV\_5,  
 NPV\_6)  
  
AVG\_NPV <- c(avgNPV\_1,  
 avgNPV\_2,  
 avgNPV\_3,  
 avgNPV\_4,  
 avgNPV\_5,  
 avgNPV\_6)  
  
df\_modified <- data.frame(Cluster, Count, Profitable, PercentProfitable, NPV, AVG\_NPV)  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 43 24 55.81395 -27862 -647.95349  
## 2 2 370 270 72.97297 13086 35.36757  
## 3 3 175 122 69.71429 -8798 -50.27429  
## 4 4 283 207 73.14488 18963 67.00707  
## 5 5 102 68 66.66667 -14593 -143.06863  
## 6 6 27 8 29.62963 -76677 -2839.88889

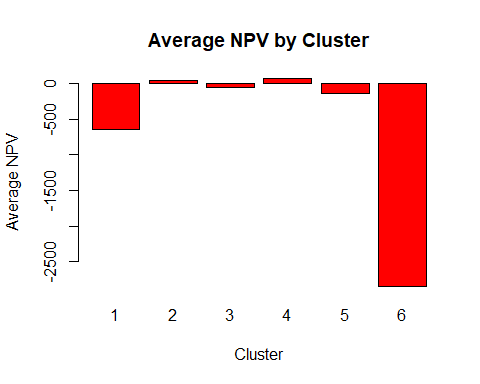
####  
barplot(PercentProfitable, names.arg = Cluster, xlab = "Cluster", ylab = "Percent Profitability", col = "blue", main = "Profitability by Cluster")



#####  
df\_modified

## Cluster Count Profitable PercentProfitable NPV AVG\_NPV  
## 1 1 43 24 55.81395 -27862 -647.95349  
## 2 2 370 270 72.97297 13086 35.36757  
## 3 3 175 122 69.71429 -8798 -50.27429  
## 4 4 283 207 73.14488 18963 67.00707  
## 5 5 102 68 66.66667 -14593 -143.06863  
## 6 6 27 8 29.62963 -76677 -2839.88889

barplot(AVG\_NPV, names.arg = Cluster, xlab = "Cluster", ylab = "Average NPV", col = "red", main = "Average NPV by Cluster")



####################  
#7  
df <- fread("Credit.csv")  
df$AMOUNT\_REQUESTED <- as.numeric(gsub(",","",df$AMOUNT\_REQUESTED))  
df$CREDIT\_EXTENDED <- NULL  
df$`OBS#` <- NULL  
df$PROFITABLE <- ifelse(df$NPV > 0, 1, 0)  
df$PROFITABLE <- factor(df$PROFITABLE)  
  
#Creating Dummy Varaibles  
df$CHK0 <- factor(ifelse(df$CHK\_ACCT == 0, 1, 0))  
df$CHK1 <- factor(ifelse(df$CHK\_ACCT == 1, 1, 0))  
df$CHK2 <- factor(ifelse(df$CHK\_ACCT == 2, 1, 0))  
df$CHK3 <- factor(ifelse(df$CHK\_ACCT == 3, 1, 0))  
  
df$SAV0 <- factor(ifelse(df$SAV\_ACCT == 0, 1, 0))  
df$SAV1 <- factor(ifelse(df$SAV\_ACCT == 1, 1, 0))  
df$SAV2 <- factor(ifelse(df$SAV\_ACCT == 2, 1, 0))  
df$SAV3 <- factor(ifelse(df$SAV\_ACCT == 3, 1, 0))  
df$SAV4 <- factor(ifelse(df$SAV\_ACCT == 4, 1, 0))  
  
df$HISTORY0 <- factor(ifelse(df$HISTORY == 0, 1, 0))  
df$HISTORY1 <- factor(ifelse(df$HISTORY == 1, 1, 0))  
df$HISTORY2 <- factor(ifelse(df$HISTORY == 2, 1, 0))  
df$HISTORY3 <- factor(ifelse(df$HISTORY == 3, 1, 0))  
df$HISTORY4 <- factor(ifelse(df$HISTORY == 4, 1, 0))  
  
df$EMPLOYMENT0 <- factor(ifelse(df$EMPLOYMENT == 0, 1, 0))  
df$EMPLOYMENT1 <- factor(ifelse(df$EMPLOYMENT == 1, 1, 0))  
df$EMPLOYMENT2 <- factor(ifelse(df$EMPLOYMENT == 2, 1, 0))  
df$EMPLOYMENT3 <- factor(ifelse(df$EMPLOYMENT == 3, 1, 0))  
df$EMPLOYMENT4 <- factor(ifelse(df$EMPLOYMENT == 4, 1, 0))  
  
df$OWN\_RES <- factor(df$OWN\_RES)  
  
df$JOB0 <- factor(ifelse(df$JOB == 0, 1, 0))  
df$JOB1 <- factor(ifelse(df$JOB == 1, 1, 0))  
df$JOB2 <- factor(ifelse(df$JOB == 2, 1, 0))  
df$JOB3 <- factor(ifelse(df$JOB == 3, 1, 0))  
  
df$CHK\_ACCT <- NULL  
df$SAV\_ACCT <- NULL  
df$HISTORY <- NULL  
df$EMPLOYMENT <- NULL  
df$JOB <- NULL  
df$NPV <- NULL  
  
df <- df[, -(1:9)]  
df <- df[, -(2:6)]  
  
set.seed(12345)  
  
rules <- apriori(df, parameter = list(supp = 0.1, conf = 0.8),   
 appearance = list(rhs = "PROFITABLE=1"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 100   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[50 item(s), 1000 transaction(s)] done [0.00s].  
## sorting and recoding items ... [42 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8

## Warning in apriori(df, parameter = list(supp = 0.1, conf = 0.8), appearance  
## = list(rhs = "PROFITABLE=1")): Mining stopped (time limit reached). Only  
## patterns up to a length of 8 returned!

## done [5.78s].  
## writing ... [264424 rule(s)] done [0.21s].  
## creating S4 object ... done [0.19s].

rules <- sort(rules, decreasing = TRUE, by = "lift")  
inspect(rules[1:20])

## lhs rhs support confidence lift count  
## [1] {CHK3=1,   
## SAV1=0,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.103 0.9716981 1.390126 103  
## [2] {CHK3=1,   
## SAV2=0,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.100 0.9708738 1.388947 100  
## [3] {CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [4] {CHK0=0,   
## CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [5] {CHK1=0,   
## CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [6] {CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB1=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [7] {CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1,   
## JOB3=0} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [8] {CHK2=0,   
## CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [9] {CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB0=0,   
## JOB2=1} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [10] {CHK3=1,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB1=0,   
## JOB3=0} => {PROFITABLE=1} 0.114 0.9661017 1.382120 114  
## [11] {CHK3=1,   
## HISTORY1=0,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.113 0.9658120 1.381705 113  
## [12] {CHK3=1,   
## HISTORY0=0,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB2=1} => {PROFITABLE=1} 0.113 0.9658120 1.381705 113  
## [13] {CHK3=1,   
## HISTORY1=0,   
## HISTORY3=0,   
## EMPLOYMENT0=0,   
## EMPLOYMENT1=0,   
## EMPLOYMENT2=0,   
## JOB1=0} => {PROFITABLE=1} 0.137 0.9647887 1.380241 137  
## [14] {OWN\_RES=1,   
## CHK3=1,   
## SAV1=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [15] {CHK0=0,   
## CHK1=0,   
## SAV1=0,   
## HISTORY1=0,   
## HISTORY3=0,   
## EMPLOYMENT4=1} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [16] {OWN\_RES=1,   
## CHK3=1,   
## SAV1=0,   
## HISTORY2=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [17] {OWN\_RES=1,   
## CHK0=0,   
## CHK3=1,   
## SAV1=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [18] {OWN\_RES=1,   
## CHK1=0,   
## CHK3=1,   
## SAV1=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [19] {OWN\_RES=1,   
## CHK3=1,   
## SAV1=0,   
## HISTORY3=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106  
## [20] {OWN\_RES=1,   
## CHK2=0,   
## CHK3=1,   
## SAV1=0,   
## HISTORY4=1,   
## EMPLOYMENT0=0} => {PROFITABLE=1} 0.106 0.9636364 1.378593 106

####################  
#9  
totalProfit = 0  
  
df\_new <- fread("Credit.csv")  
  
for (i in 1:nrow(df\_new)) {  
 if(df[i]$CHK3 == 1 & df[i]$SAV1 == 0 & df[i]$HISTORY3 == 0 & df[i]$EMPLOYMENT0 == 0 & df[i]$EMPLOYMENT1 == 0 & df[i]$EMPLOYMENT2 == 0 & df[i]$JOB2 == 1)  
 totalProfit = totalProfit + df\_new[i]$NPV  
}  
  
totalProfit

## [1] 40824

avgProfit <- totalProfit/nrow(df\_new)  
avgProfit

## [1] 40.824

###########################  
#10  
credit <- read\_csv("Credit.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## AMOUNT\_REQUESTED = col\_number(),  
## CREDIT\_EXTENDED = col\_number()  
## )

## See spec(...) for full column specifications.

credit$PROFITABLE <- as.factor(ifelse(credit$NPV>0,1,0))  
  
set.seed(12345)  
km.out <- kmeans(credit[-16:-17], 5, nstart=20)  
  
  
credit$CLUSTER <- km.out$cluster  
  
credit.sub <- subset(credit, select = c("PROFITABLE","CLUSTER"))  
  
credit.sub$CLUSTER\_1 <- ifelse(credit.sub$CLUSTER != 1, 0, 1)  
credit.sub$CLUSTER\_2 <- ifelse(credit.sub$CLUSTER != 2, 0, 1)  
credit.sub$CLUSTER\_3 <- ifelse(credit.sub$CLUSTER != 3, 0, 1)  
credit.sub$CLUSTER\_4 <- ifelse(credit.sub$CLUSTER != 4, 0, 1)  
credit.sub$CLUSTER\_5 <- ifelse(credit.sub$CLUSTER != 5, 0, 1)  
credit.sub$CLUSTER <- NULL  
  
credit.sub[] <- lapply(credit.sub, factor)  
col\_names <- names(credit.sub)  
  
set.seed(12345)  
rules<-apriori(data=credit.sub, parameter=list(supp=0.01, conf=0.08),   
 appearance = list(default="lhs",rhs="PROFITABLE=1"),  
 control = list(verbose=F))  
  
  
rules<-sort(rules, decreasing=TRUE,by="lift")  
inspect(rules[1:20])

## lhs rhs support confidence lift count  
## [1] {CLUSTER\_1=1} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [2] {CLUSTER\_1=1,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [3] {CLUSTER\_1=1,   
## CLUSTER\_3=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [4] {CLUSTER\_1=1,   
## CLUSTER\_2=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [5] {CLUSTER\_1=1,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [6] {CLUSTER\_1=1,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [7] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [8] {CLUSTER\_1=1,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [9] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_3=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [10] {CLUSTER\_1=1,   
## CLUSTER\_3=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [11] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [12] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [13] {CLUSTER\_1=1,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [14] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [15] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_3=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [16] {CLUSTER\_2=0,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [17] {CLUSTER\_1=1,   
## CLUSTER\_2=0,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.092 1.0000000 1.430615 92  
## [18] {CLUSTER\_2=0,   
## CLUSTER\_4=0,   
## CLUSTER\_5=0} => {PROFITABLE=1} 0.312 0.8167539 1.168461 312  
## [19] {CLUSTER\_2=0,   
## CLUSTER\_3=0,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.098 0.8032787 1.149183 98  
## [20] {CLUSTER\_2=0,   
## CLUSTER\_4=0} => {PROFITABLE=1} 0.318 0.7718447 1.104213 318

topRule <- subset(credit, CLUSTER == 1)  
  
totalProfits <- sum(topRule$NPV)  
totalProfit

## [1] 40824

averageProfits <- mean(topRule$NPV)  
averageProfits

## [1] 1168.228