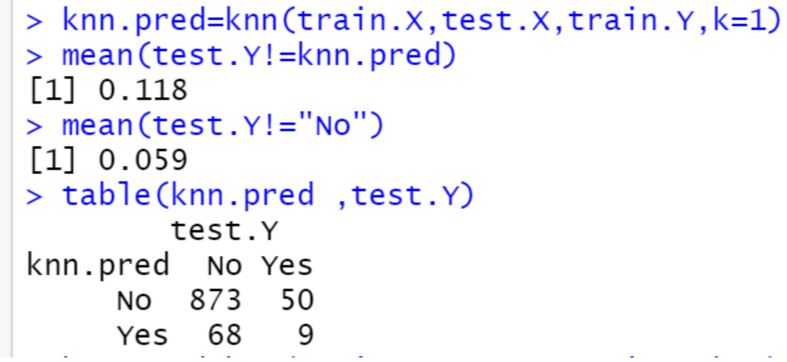
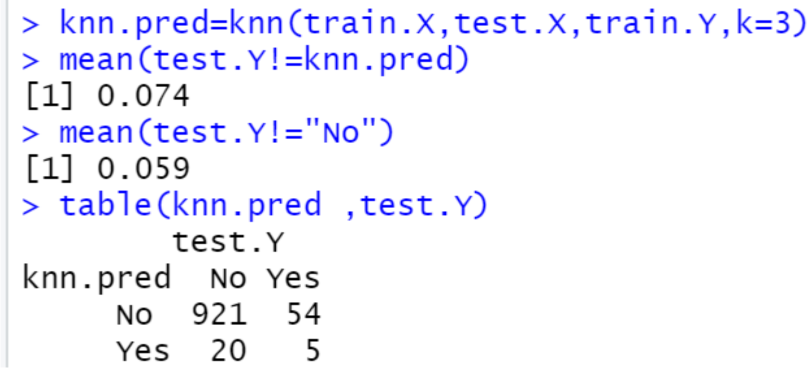
Appendix:

Table1: KNN output when k=1



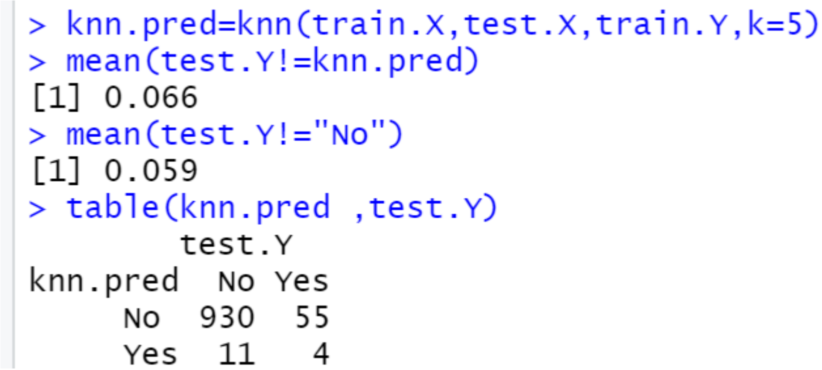
```
> knn.pred=knn(train.X,test.X,train.Y,k=1)
> mean(test.Y!=knn.pred)
[1] 0.118
> mean(test.Y!="No")
[1] 0.059
> table(knn.pred ,test.Y)
        test.Y
knn.pred  No Yes
     No  873  50
     Yes  68   9
```

Table2: KNN output when k=3



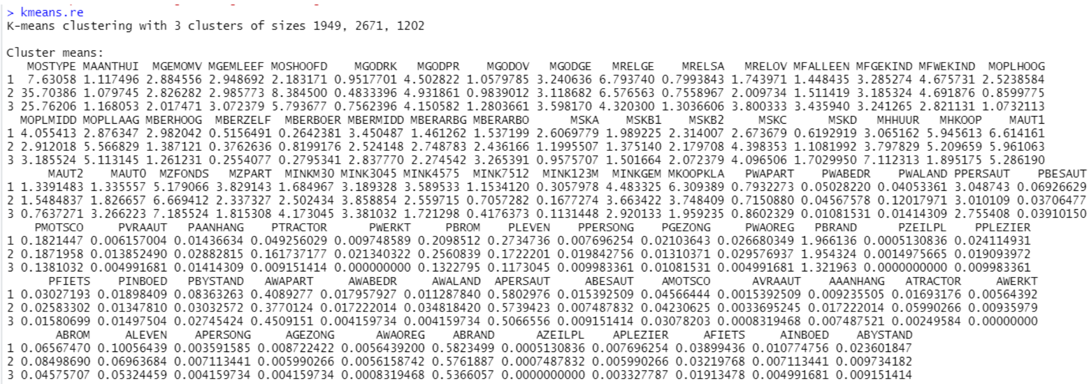
```
> knn.pred=knn(train.X,test.X,train.Y,k=3)
> mean(test.Y!=knn.pred)
[1] 0.074
> mean(test.Y!="No")
[1] 0.059
> table(knn.pred ,test.Y)
        test.Y
knn.pred  No Yes
     No  921  54
     Yes  20   5
```

Table3: KNN output when k=5



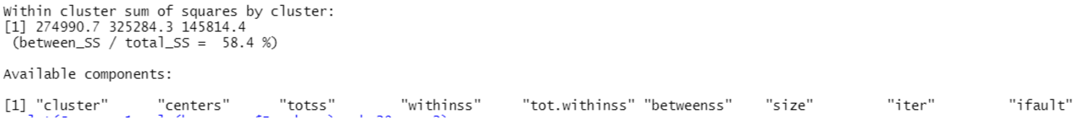
```
> knn.pred=knn(train.X,test.X,train.Y,k=5)
> mean(test.Y!=knn.pred)
[1] 0.066
> mean(test.Y!="No")
[1] 0.059
> table(knn.pred ,test.Y)
        test.Y
knn.pred  No Yes
     No  930  55
     Yes  11   4
```

Table4: output of kmeans.re



```
> kmeans.re
K-means clustering with 3 clusters of sizes 1949, 2671, 1202

Cluster means:
   MOSTYPE MAANTHUI  MGEMOMV MGEMLEEF MOSHOOFD    MGODRK   MGODPR    MGODOV   MGODGE   MRELGE    MRELSA   MRELOV MFALLEEN MFGEKIND MFWEKIND  MOPLHOOG
1  7.63058 1.117496 2.884556 2.948692 2.183171 0.9517701 4.502822 1.0579785 3.240636 6.793740 0.7993843 1.743971 1.448435 3.285274 4.675731 2.5238584
2 35.70386 1.079745 2.826282 2.985773 8.384500 0.4833396 4.931861 0.9839012 3.118682 6.576563 0.7558967 2.009734 1.511419 3.185324 4.691876 0.8599775
3 25.76206 1.168053 2.017471 3.072379 5.793677 0.7562396 4.150582 1.2803661 3.598170 4.320300 1.3036606 3.800333 3.435940 3.241265 2.821131 1.0732113
  MOPLMIDD MOPLLAAG MBERHOOG  MBERZELF  MBERBOER MBERMIDD MBERARBG MBERARBO      MSKA    MSKB1    MSKB2     MSKC      MSKD   MHHUUR   MHKOOP    MAUT1
1 4.055413 2.876347 2.982042 0.5156491 0.2642381 3.450487 1.461262 1.537199 2.6069779 1.989225 2.314007 2.673679 0.6192919 3.065162 5.945613 6.614161
2 2.912018 5.566829 1.387121 0.3762636 0.8199176 2.524148 2.748783 2.436166 1.1995507 1.375140 2.179708 4.398353 1.1081992 3.797829 5.209659 5.961063
3 3.185524 5.113145 1.261231 0.2554077 0.2795341 2.837770 2.274542 3.265391 0.9575707 1.501664 2.072379 4.096506 1.7029950 7.112313 1.895175 5.286190
      MAUT2    MAUT0  MZFONDS    MZPART  MINKM30 MINK3045 MINK4575  MINK7512  MINK123M  MINKGEM MKOOPKLA   PWAPART    PWABEDR    PWALAND PPERSAUT    PBESAUT
1 1.3391483 1.335557 5.179066 3.829143 1.684967 3.189328 3.589533 1.1534120 0.3057978 4.483325 6.309389 0.7932273 0.05028220 0.04053361 3.048743 0.06926629
2 1.5484837 1.826657 6.669412 2.337327 2.502434 3.858854 2.559715 0.7057282 0.1677274 3.663422 3.748409 0.7150880 0.04567578 0.12017971 3.010109 0.03706477
3 0.7637271 3.266223 7.185524 1.815308 4.173045 3.381032 1.721298 0.4176373 0.1131448 2.920133 1.959235 0.8602329 0.01081531 0.01414309 2.755408 0.03910150
    PMOTSCO     PVRAAUT   PAANHANG    PTRACTOR      PWERKT     PBROM    PLEVEN    PPERSONG    PGEZONG     PWAOREG   PBRAND      PZEILPL    PPLEZIER
1 0.1821447 0.006157004 0.01436634 0.049256029 0.009748589 0.2098512 0.2734736 0.007696254 0.02103643 0.026680349 1.966136 0.0005130836 0.024114931
2 0.1871958 0.013852490 0.02882815 0.161737177 0.021340322 0.2560839 0.1722201 0.019842756 0.01310371 0.029576937 1.954324 0.0014975665 0.019093972
3 0.1381032 0.004991681 0.01414309 0.009151414 0.000000000 0.1322795 0.1173045 0.009983361 0.01081531 0.004991681 1.321963 0.0000000000 0.009983361
      PFIETS    PINBOED   PBYSTAND   AWAPART     AWABEDR     AWALAND  APERSAUT     ABESAUT    AMOTSCO      AVRAAUT     AAANHANG   ATRACTOR     AWERKT
1 0.03027193 0.01898409 0.08363263 0.4089277 0.017957927 0.011287840 0.5802976 0.015392509 0.04566444 0.0015392509 0.009235505 0.01693176 0.00564392
2 0.02583302 0.01347810 0.03032572 0.3770124 0.017222014 0.034818420 0.5739423 0.007487832 0.04230625 0.0033695245 0.017222014 0.05990266 0.00935979
3 0.01580699 0.01497504 0.02745424 0.4509151 0.004159734 0.004159734 0.5066556 0.009151414 0.03078203 0.0008319468 0.007487521 0.00249584 0.00000000
       ABROM     ALEVEN    APERSONG     AGEZONG      AWAOREG    ABRAND      AZEILPL   APLEZIER     AFIETS     AINBOED    ABYSTAND
1 0.06567470 0.10056439 0.003591585 0.008722422 0.0056439200 0.5823499 0.0005130836 0.007696254 0.03899436 0.010774756 0.023601847
2 0.08498690 0.06963684 0.007113441 0.005990266 0.0056158742 0.5761887 0.0007487832 0.005990266 0.03219768 0.007113441 0.009734182
3 0.04575707 0.05324459 0.004159734 0.004159734 0.0008319468 0.5366057 0.0000000000 0.003327787 0.01913478 0.004991681 0.009151414
```



```
Within cluster sum of squares by cluster:
[1] 274990.7 325284.3 145814.4
 (between_SS / total_SS =  58.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
```

Figure1: K-means clustering plot 1

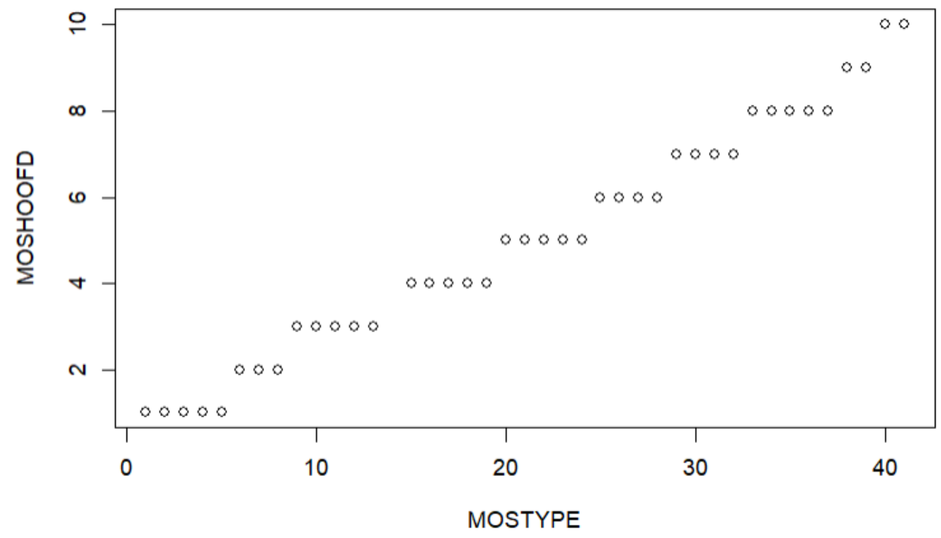


Figure2: K-means clustering plot 2

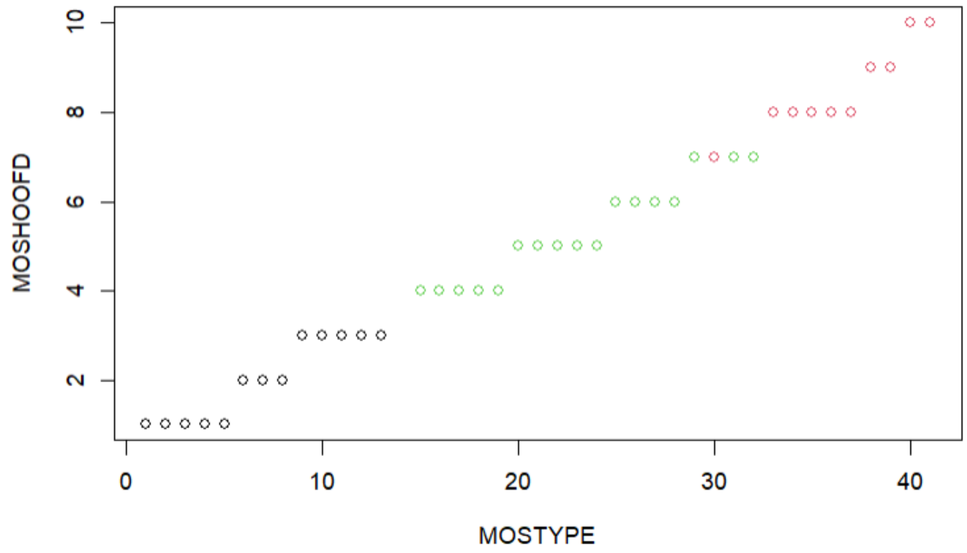


Figure3: K-means clustering with 3 clusters final step

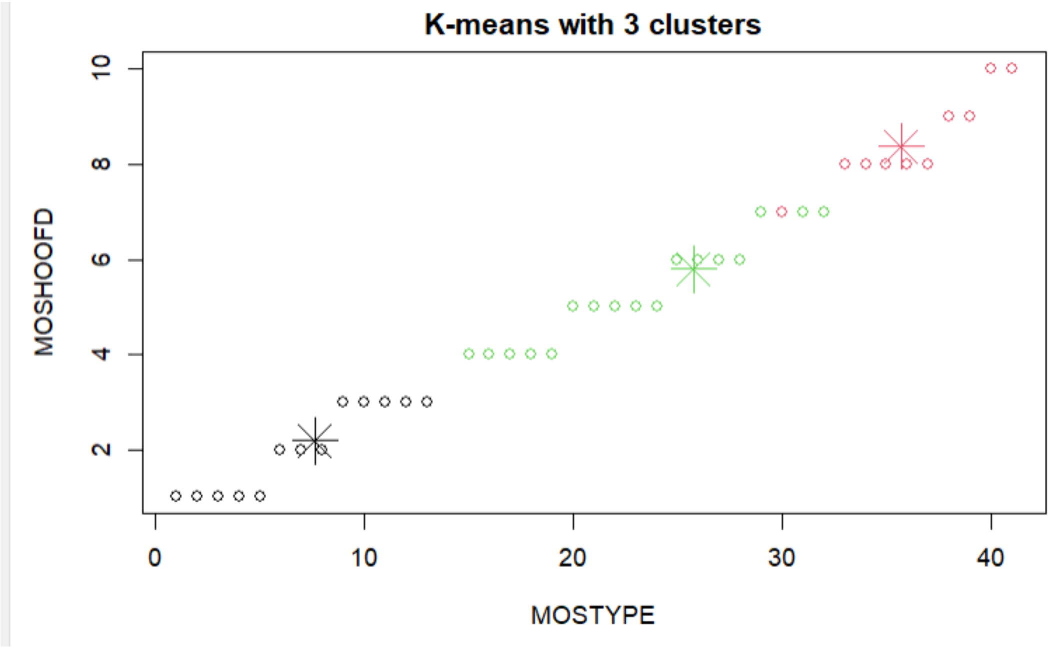


Figure4: Cluster caravan

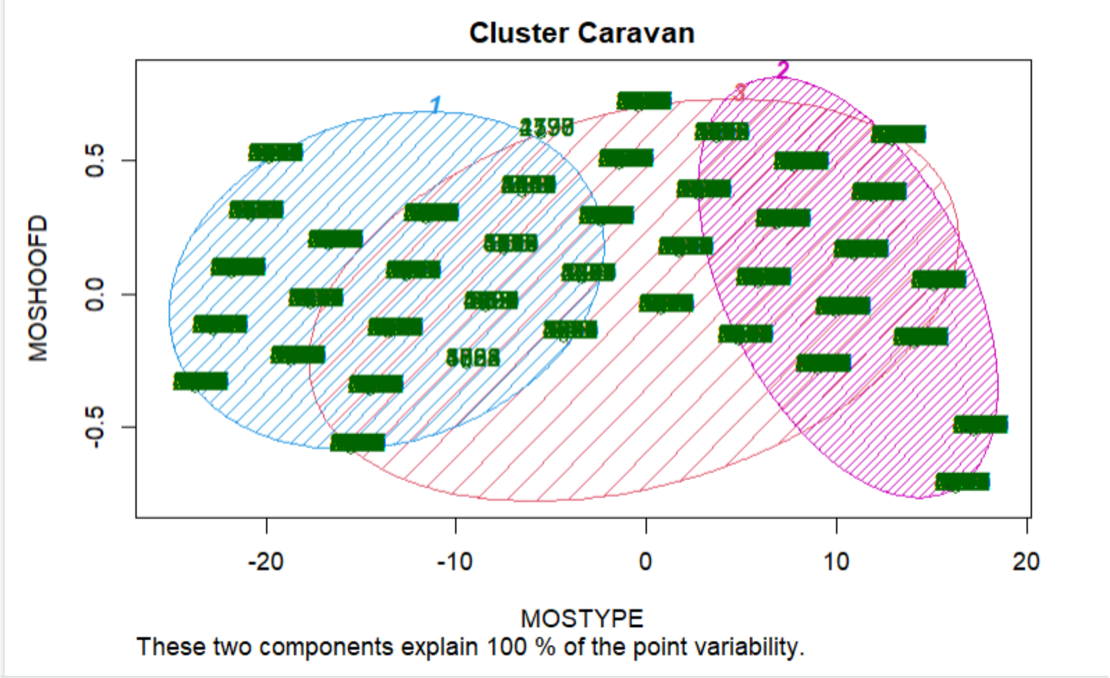
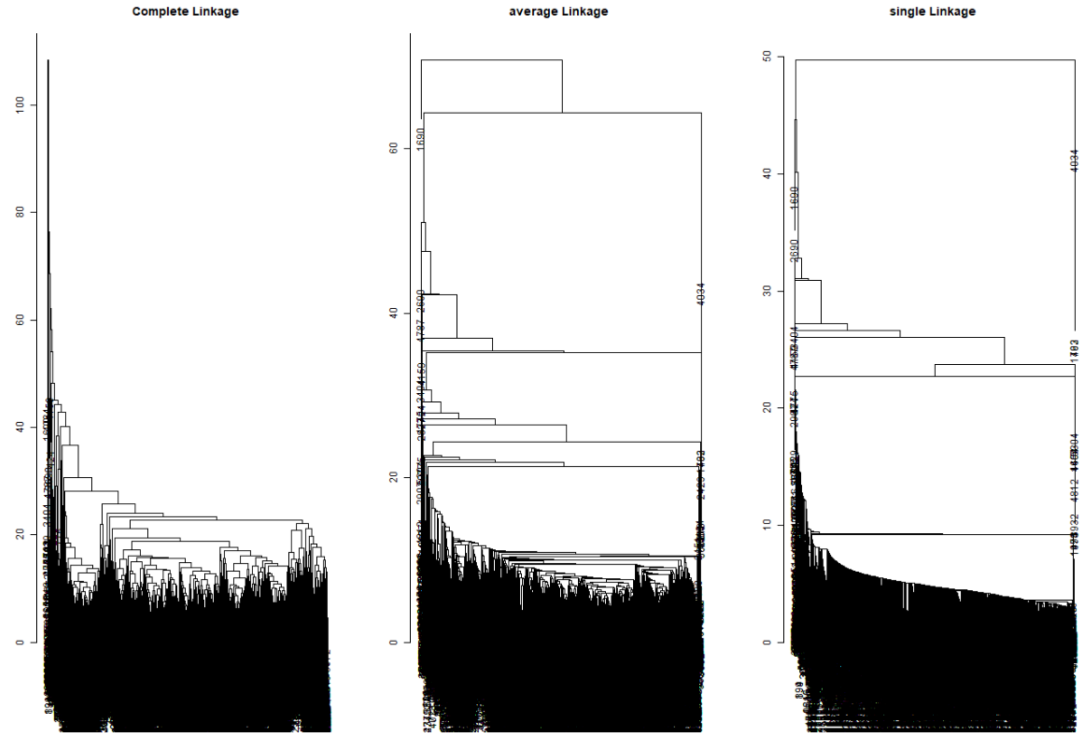


Figure5: output of Hierarchical by complete, average and single linkage.



Code for KNN:

```
library(ISLR)
library(class)#KNN
data(Caravan)
set.seed(1)

dim(Caravan)
attach(Caravan)
summary(Purchase)
#0.0598 people purchased caravan insurance

#standardize the data so that all variables are given a mean of zero and a standard deviation
of one
#then all variables will be on a   scale
#the column 86 is the qualitative purchase variable
standardized.X=scale(Caravan[,-86])
var(Caravan[ ,1])
var(Caravan[ ,2])
var(standardized.X[,1])
var(standardized.X[,2])

#split6 the observations into a test set, containing the first 1000 observations,
#and the training set containing the remaining observations
test=1:1000
train.X= standardized.X[-test ,]
test.X= standardized.X[test ,]
train.Y=Purchase [-test]
test.Y=Purchase [test]

#fit KNN model on the training data using k=1
knn.pred=knn(train.X,test.X,train.Y,k=2)
mean(test.Y!=knn.pred)
mean(test.Y!="No")
table(knn.pred ,test.Y)

knn.pred=knn(train.X,test.X,train.Y,k=3)
mean(test.Y!=knn.pred)
mean(test.Y!="No")
table(knn.pred ,test.Y)

knn.pred=knn(train.X,test.X,train.Y,k=5)
mean(test.Y!=knn.pred)
mean(test.Y!="No")
table(knn.pred ,test.Y)
```

code for K-means:

```
# Installing Packages
install.packages("ClusterR")
install.packages("cluster")

# Loading package
library(ClusterR)
library(cluster)
library(ISLR)
data(Caravan)
head(Caravan)
# Removing initial label of
# Species from original dataset
Caravan_1 <- Caravan[, -86]#remove the response variable the 86th column

# Fitting K-Means clustering Model
# to training dataset
set.seed(240) # Setting seed
kmeans.re <- kmeans(Caravan_1, centers = 3, nstart = 20)#the x's variable, centers normally set to 3, from the 20th begin***
kmeans.re

# Cluster identification for
# each observation
kmeans.re$cluster

# Confusion Matrix
cm <- table(Caravan$Purchase, kmeans.re$cluster)
cm    #test the accurecy

# Model Evaluation and visualization
plot(Caravan_1[c("MOSTYPE", "MOSHOOFD")])
plot(Caravan_1[c("MOSTYPE", "MOSHOOFD")],
        col = kmeans.re$cluster)
plot(Caravan_1[c("MOSTYPE", "MOSHOOFD")],
        col = kmeans.re$cluster,
        main = "K-means with 3 clusters")

## Plotiing cluster centers
kmeans.re$centers
kmeans.re$centers[, c("MOSTYPE", "MOSHOOFD")]

# cex is font size, pch is symbol
points(kmeans.re$centers[, c("MOSTYPE", "MOSHOOFD")],
       col = 1:3, pch = 8, cex = 3)

## Visualizing clusters
y_kmeans <- kmeans.re$cluster
clusplot(Caravan_1[, c("MOSTYPE", "MOSHOOFD")],
         y_kmeans,
         lines = 0,
         shade = TRUE,
         color = TRUE,
         labels = 2,
         plotchar = FALSE,
         span = TRUE,
         main = paste("Cluster Caravan"),
         xlab = 'MOSTYPE',
         ylab = 'MOSHOOFD')
```