```
# Read in the Titanic dataset. We will work with a subset.
myData = cbind(titanic[,"pclass"],titanic[,"survived"],titanic[,"sex"],titanic[,"age"])#pull out the columns
dim(myData)
summary(myData)
myDataClean = na.omit((myData))
dim(myDataClean)
myDataClean=as.data.frame(myDataClean)# class(myDataClean) is matrix but need data.frame
myDataClean[,2]=as.factor(myDataClean[,2])
myDataClean[,3]=as.factor(myDataClean[,3])
myDataClean[,1]=as.factor(myDataClean[,1])
myDataClean[,4] = ifelse(myDataClean[,4]<18,"child", "adult")
myDataClean[,4]=as.factor(myDataClean[,4])
summary(myDataClean)
colnames(myDataClean) = c("Class","Survived","Sex","Age")
summary(myDataClean)

# First we'll use model.matrix to convert the factors.
# We need myDataClean to be a data frame to do this ...
myDataClean = as.data.frame(myDataClean)
titanticBinaryMM = data.frame(model.matrix(~., data=myDataClean)[,-1])
head(titanticBinaryMM)

# Only includes the dummy variables - will use to cluster records
# Now let's create some distance matrices
titanicDistBinaryMM = dist(titanticBinaryMM, method = "binary")
titanicDistEuclMM = dist(titanticBinaryMM, method = "euclidian")
fitMMBi = hclust(d=titanicDistBinaryMM,method="ward.D2")
fitMMEu = hclust(d=titanicDistEuclMM,method="ward.D2")
plot(fitMMBi, labels = FALSE)
xMMbi=cutree(fitMMBi,3)
y=myDataClean[xMMbi == 1,]
summary(y)
y=myDataClean[xMMbi == 2,]
summary(y)
y=myDataClean[xMMbi == 3,]
summary(y)
xMMEu=cutree(fitMMEu,3)
y=myDataClean[xMMEu == 1,]
summary(y)
y=myDataClean[xMMEu == 2,]
summary(y)
y=myDataClean[xMMEu == 3,]
summary(y)
table(xMMbi,xMMEu)

# Pretty close, somewhat different cluster for first group
# Now let's try the ade4 conversion of the factors
# Attach the "ade4"package
newBinaryData = acm.disjonctif(myDataClean)
head(newBinaryData)

# Variable for each factor
# We'll do the binary and Euclidian distances again
titanicDistBinaryade4 = dist(newBinaryData, method = "binary")
titanicDistEuclade4 = dist(newBinaryData, method = "euclidian")
fitade4Bi = hclust(d=titanicDistBinaryade4,method="ward.D2")
fitade4Eu = hclust(d=titanicDistEuclade4,method="ward.D2")
xade4bi=cutree(fitade4Bi,3)
y=myDataClean[xade4bi == 1,]
summary(y)
y=myDataClean[xade4bi == 2,]
summary(y)
y=myDataClean[xade4bi == 3,]
summary(y)
xade4eu=cutree(fitade4Eu,3)
y=myDataClean[xade4eu == 1,]
summary(y)
y=myDataClean[xade4eu == 2,]
```

```r
summary(y)
y=myDataClean[xade4eu == 3,]
summary(y)
table(xade4bi,xade4eu)

# Now very close alignment - seems to suggest using ade4 may be
# better? We can also compare to K-modes clustering ...
# Attach the "klaR" package
km = kmodes(myDataClean, 3)

# The clustering is differnet again ...
names(km)
y=myDataClean[km$cluster == 1,]
summary(y)
table(xade4bi,km$cluster)


# Read in the Adult Data dataset
myData = cbind(adultData$age,adultData$relationship,adultData$race,adultData$sex,adultData$income)
myData = as.data.frame(myData)
colnames(myData)=c("Age","Relationship","Race","Sex","Income")
summary(myData)
myData$Relationship=as.factor(myData$Relationship)
myData$Race=as.factor(myData$Race)
myData$Sex=as.factor(myData$Sex)
myData$Income=as.factor(myData$Income)
myDataShort = myData[sample(nrow(myData),1000),]

# We'll first use model.matrix to convert the matrix, and then
# scale it
adultMM = data.frame(model.matrix(~., data=myDataShort)[,-1])# for biased -1
head(adultMM)
adultMM = scale(adultMM)

# Now we do a Euclidian metric to get the distances
adultDistEuclMM = dist(adultMM, method = "euclidian")

# Cluster using Hclust
fitMMEucl = hclust(d=adultDistEuclMM,method="ward.D2")
plot(fitMMEucl, labels = FALSE)

# Only two groups?
xMMEucl=cutree(fitMMEucl,2)
y=myDataShort[xMMEucl == 1,]
summary(y)
y=myDataShort[xMMEucl == 2,]
summary(y)

# Now let's try daisy using the Gower distance on the original
# matrix. Attach the "cluster" package
AdultDist = daisy(myDataShort)
fit <- hclust(d=AdultDist, method="ward.D2")
plot(fit, labels=FALSE)

# Better clustering
# Let's look at 4 clusters
xDaisy=cutree(fit,4)
y=myDataShort[xDaisy == 1,]
summary(y)
y=myDataShort[xDaisy == 2,]
summary(y)
y=myDataShort[xDaisy == 3,]
summary(y)
y=myDataShort[xDaisy == 4,]
summary(y)
# Let's see if Euclidian was close with a 4-cut ...
xMMEucl=cutree(fitMMEucl,4)
table(xMMEucl,xDaisy)
```