R Notebook

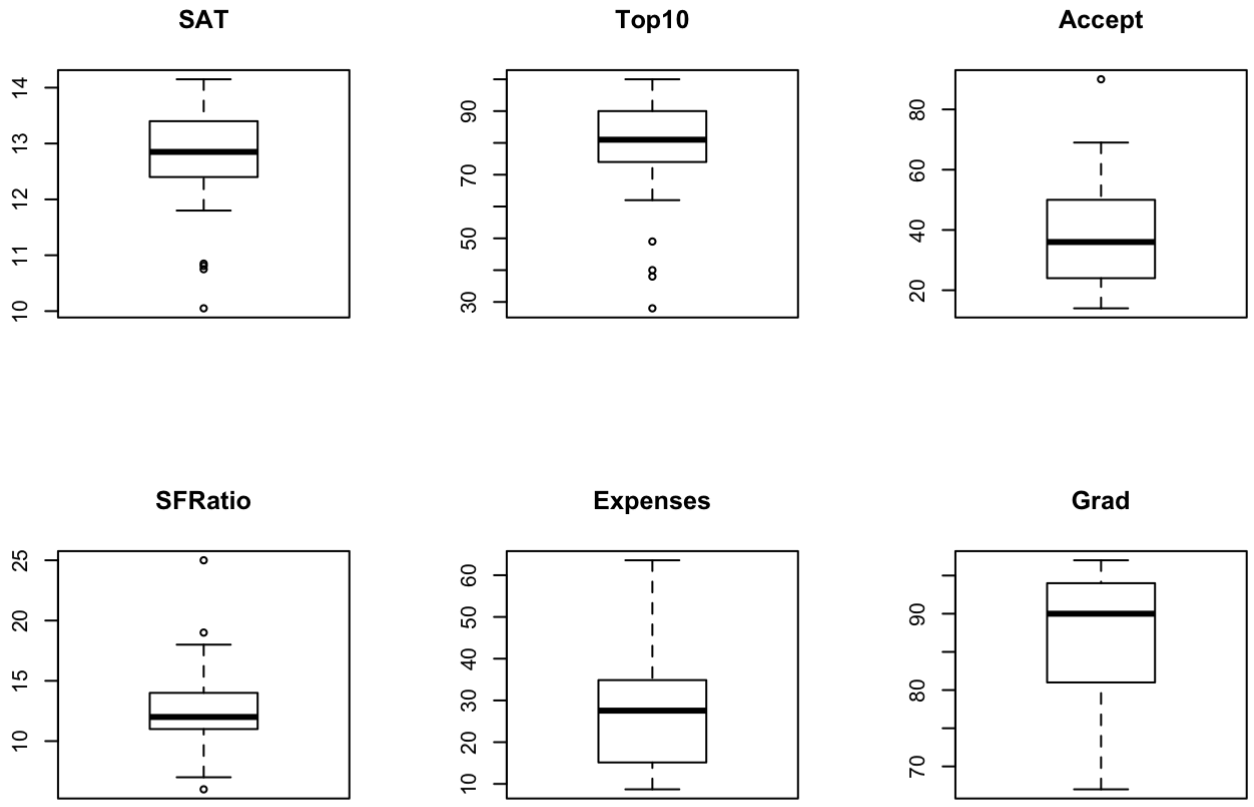
The goal of the data analysis is to cluster American universities, and see if there are any interesting groups that come out which can provide interesting insight into similarities between universities. The dataset contains the following characteristics: SAT score, tuition fee, acceptance rate, percent of class graduated, etc.

First, I checked for outliers and performed transformations to see if they could be reduced. Upon squaring the variables, there is a considerable decrease in the number of outliers. After the transformation, the variables were standardized.

```
par(mfrow=c(2,3))
UNIV <- read.csv("Universities.csv", as.is = TRUE)
library(fpc)
library(cluster)
library(ape)
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.5.2
```

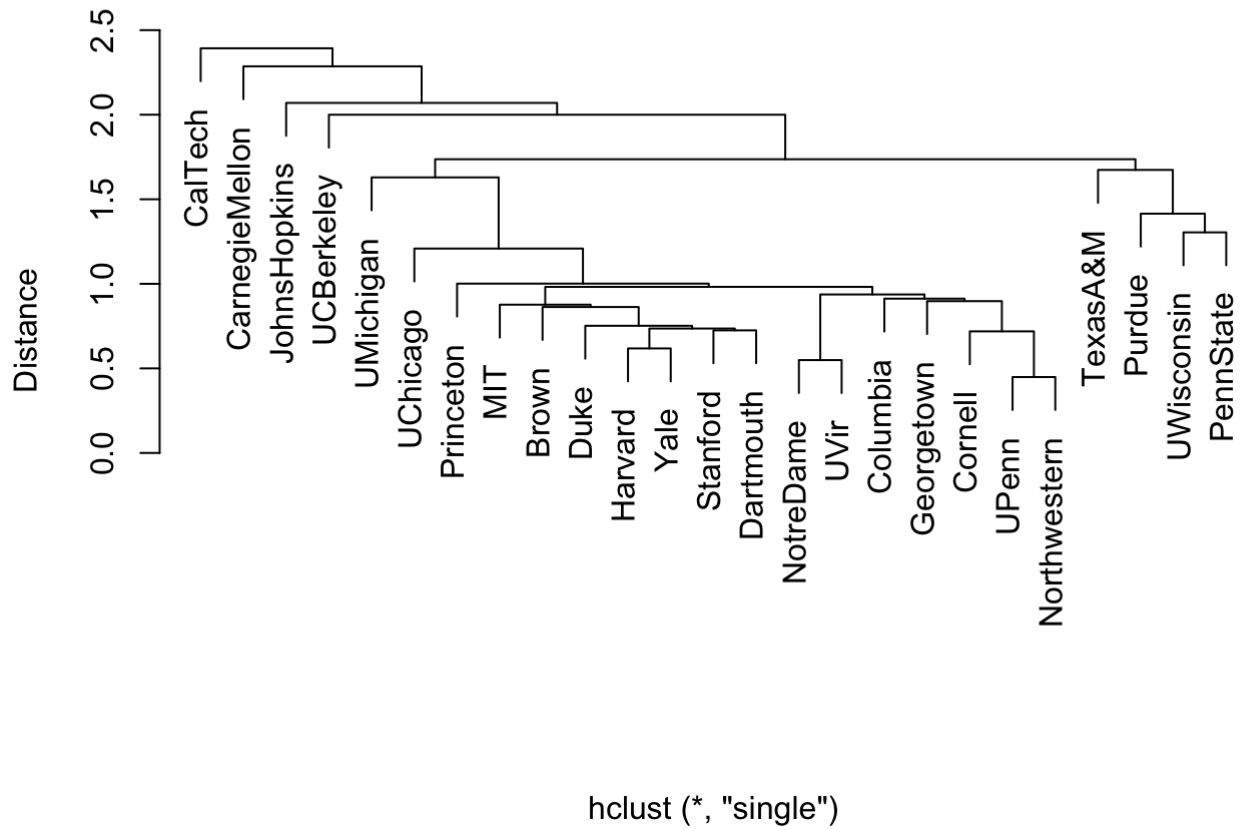
```
for (i in 2:7){
  boxplot(UNIV[,i], main = colnames(UNIV)[i])
}
```



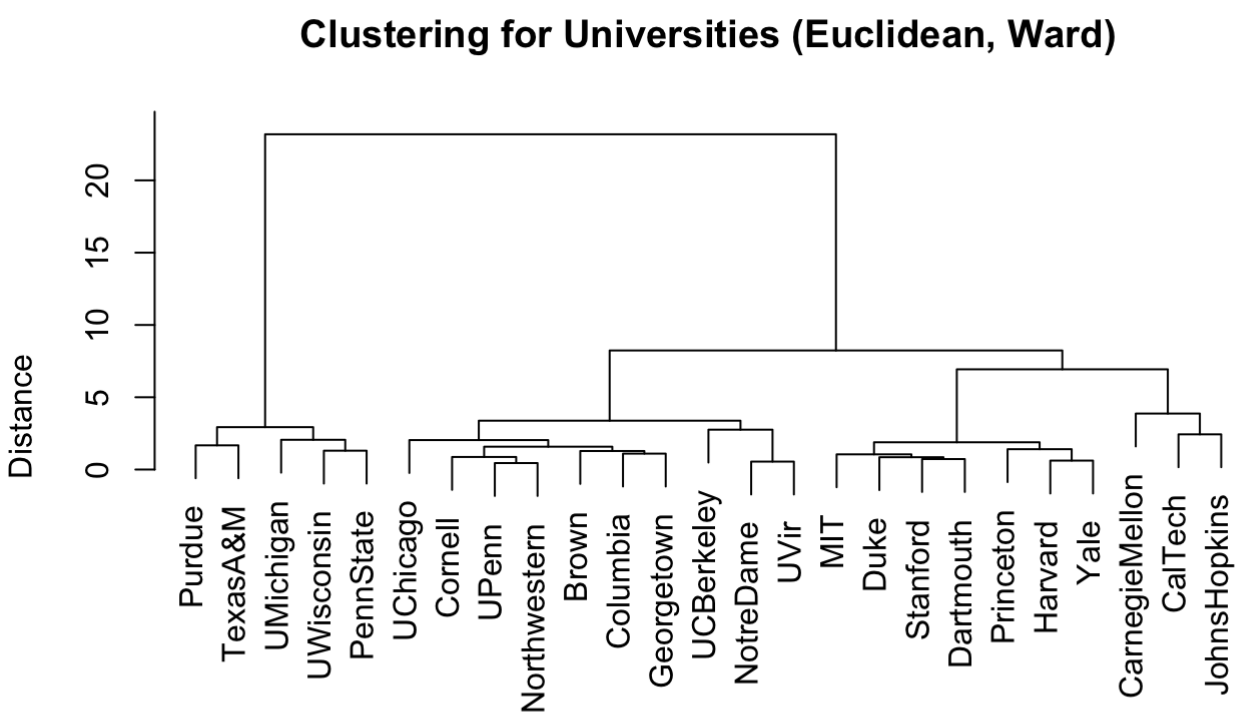
```
UNIVnormal <- UNIV
UNIVnormal[,c(2,3,7)] <- (UNIVnormal[,c(2,3,7)])^2
UNIVnormal[,c(4,5,6)] <- sqrt(UNIVnormal[,c(4,5,6)])
rownames(UNIVnormal) <- UNIVnormal[,1]
UNIVnormal <- UNIVnormal[,c(2:7)]

for (i in 1:6){
  UNIVnormal[,i] <- (UNIVnormal[,i] - mean(UNIVnormal[,i]))/sqrt(var(UNIVnormal[,i]))
}
```

Since the variables being used are continuous, Euclidean distance is an appropriate measure for the clustering, since it accurately captures numeric distances between two schools in terms of their characteristics (as opposed to say Manhattan distance, since adding the difference in tuition to the difference in admission rates does not make sense). I used both single (nearest neighbor) and Ward methods. Single aggregation would prevent outliers in clusters from being accentuated, which is an issue to be wary of given that there were outliers present in the boxplots. Ward aggregation on the other hand measures the difference between the variance within a joint cluster 1 and 2 and the sum of their individual variances. If the difference was large (i.e. Ward metric is high), this means there is much more variance when we consider the two clusters as one, and so we would like to separate them. Conversely, if the difference was low (i.e. Ward metric is low), combining the clusters would be no different from separating them, and so we consider the clusters to be close to one another.

```
par(mfrow=c(1,1))
univDistances1 <- dist(UNIVnormal[,c(1:6)], method="euclidean")

univClusters1a <- hclust(univDistances1, method = "single")
univClusters1b <- hclust(univDistances1, method = "ward.D")

univClusters1a$labels <- rownames(UNIVnormal)
univClusters1b$labels <- rownames(UNIVnormal)

plot(univClusters1a, xlab="", ylab="Distance", main="Clustering for Universities (Euclid
ean, Complete)")
```




```
plot(univClusters1b, xlab="", ylab="Distance", main="Clustering for Universities (Euclid
ean, Ward)")
```



The results group Purdue, TexasA&M, PennState, and UWisconsin together, which is expected when accounting for factors such as admission rate and tuition fee. It is interesting that amongst the Ivy League, the Ward clustering separates Cornell, UPenn, Brown, and Columbia from Harvard, Yale, Princeton and Dartmouth, and that Stanford and MIT are put with the latter group - which attests to the rise these STEM schools in the recent years, in comparison to Carnegie Mellon, CalTech, and Johns Hopkins which have been clustered into a separate group.

Using kmeans with 4 clusters, similar groups are observed, although now the group with Cornell, UPenn, Columbia has been merged with the group containing UChicago, Georgetown, NotreDame, University of Virginia, etc.

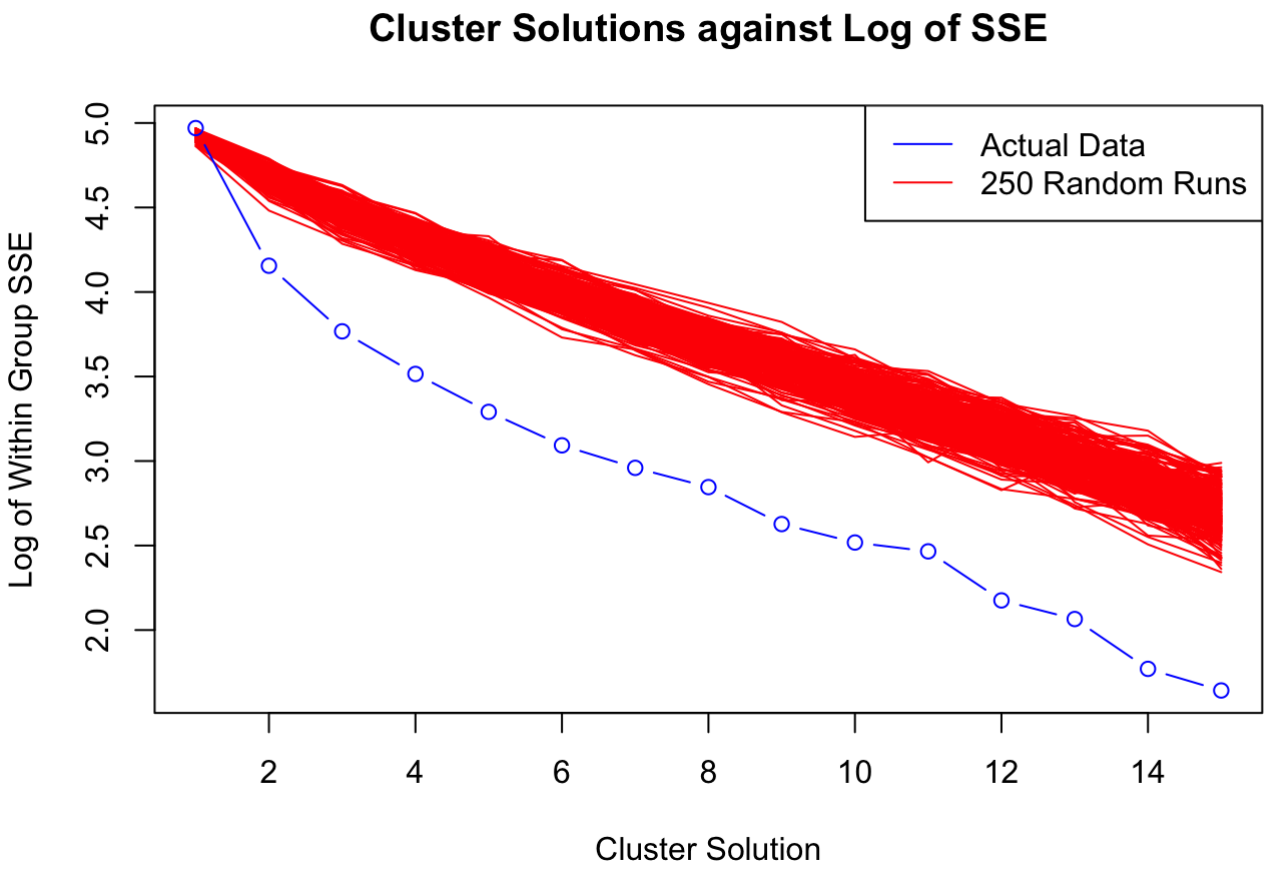
```
univKMEAN <- kmeans(UNIVnormal, centers = 4)
#see which countries are in each cluster
for (i in 1:4){
  print(paste("Schools in Cluster #",i))
  print(univKMEAN$cluster[univKMEAN$cluster==i])
}
```

```
## [1] "Schools in Cluster # 1"
##         Duke        Brown        UPenn      Cornell Northwestern
##            1            1            1            1            1
##     Columbia    NotreDame         UVir   Georgetown   UCBerkeley
##            1            1            1            1            1
## [1] "Schools in Cluster # 2"
##   JohnsHopkins       UChicago CarnegieMellon
##              2              2              2
## [1] "Schools in Cluster # 3"
##  UMichigan UWisconsin  PennState     Purdue   TexasA&M
##          3          3          3          3          3
## [1] "Schools in Cluster # 4"
##   Harvard Princeton      Yale  Stanford       MIT   CalTech Dartmouth
##         4         4         4         4         4         4         4
```

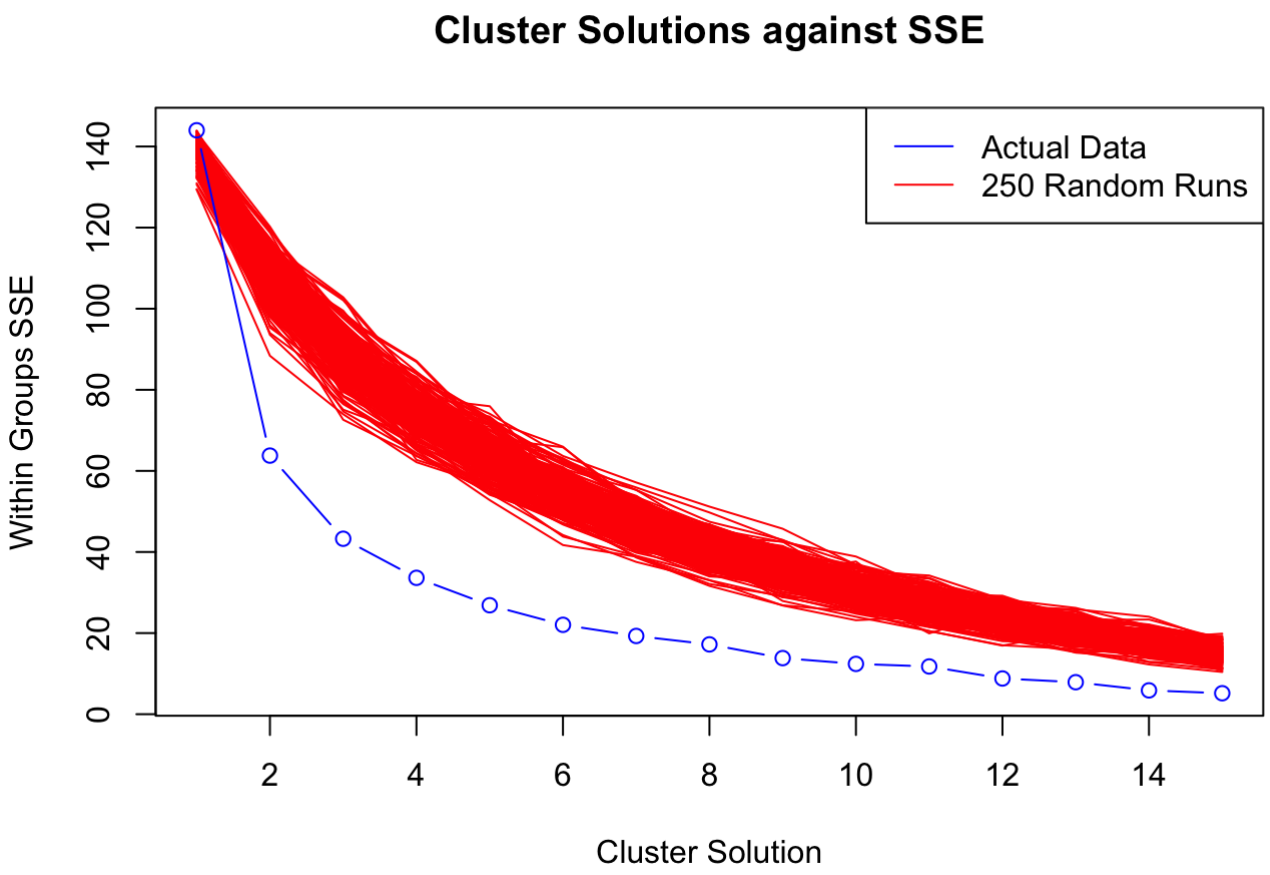
We then plot the total within group variance generated from kmeans in comparison to 250 random groupings. As expected, the variance decreases with increase in number of clusters, since more homogenous groups containing fewer elements are produced, which naturally leads to a lower variance. The kmeans cluster's variance is consistently lower than that of a randomly generated clustering across all number of clusters, which proves that kmeans successfully generates relatively homogenous clusters.

```
#kdata is just normalized input dataset
UNIVnormal <- scale(na.omit(UNIVnormal))
kdata <- UNIVnormal
n.lev <- 15  #set max value for number of clusters k

# Calculate the within groups sum of squared error (SSE) for the number of cluster solut
ions selected by the user
wss <- rnorm(10)
while (prod(wss==sort(wss,decreasing=T))==0) {
  wss <- (nrow(kdata)-1)*sum(apply(kdata,2,var))
  for (i in 2:n.lev) wss[i] <- sum(kmeans(kdata, centers=i)$withinss)}

# Calculate the within groups SSE for 250 randomized data sets (based on the original in
put data)
k.rand <- function(x){
  km.rand <- matrix(sample(x),dim(x)[1],dim(x)[2])
  rand.wss <- as.matrix(dim(x)[1]-1)*sum(apply(km.rand,2,var))
  for (i in 2:n.lev) rand.wss[i] <- sum(kmeans(km.rand, centers=i)$withinss)
  rand.wss <- as.matrix(rand.wss)
  return(rand.wss)
}

rand.mat <- matrix(0,n.lev,250)

k.1 <- function(x) {
  for (i in 1:250) {
    r.mat <- as.matrix(suppressWarnings(k.rand(kdata)))
    rand.mat[,i] <- r.mat}
  return(rand.mat)
}

# Same function as above for data with < 3 column variables
k.2.rand <- function(x){
  rand.mat <- matrix(0,n.lev,250)
  km.rand <- matrix(sample(x),dim(x)[1],dim(x)[2])
  rand.wss <- as.matrix(dim(x)[1]-1)*sum(apply(km.rand,2,var))
  for (i in 2:n.lev) rand.wss[i] <- sum(kmeans(km.rand, centers=i)$withinss)
  rand.wss <- as.matrix(rand.wss)
  return(rand.wss)
}

k.2 <- function(x){
  for (i in 1:250) {
    r.1 <- k.2.rand(kdata)
    rand.mat[,i] <- r.1}
  return(rand.mat)
}

# Determine if the data data table has > or < 3 variables and call appropriate function
 above
if (dim(kdata)[2] == 2){
  rand.mat <- k.2(kdata)}else{
  rand.mat <- k.1(kdata)
```

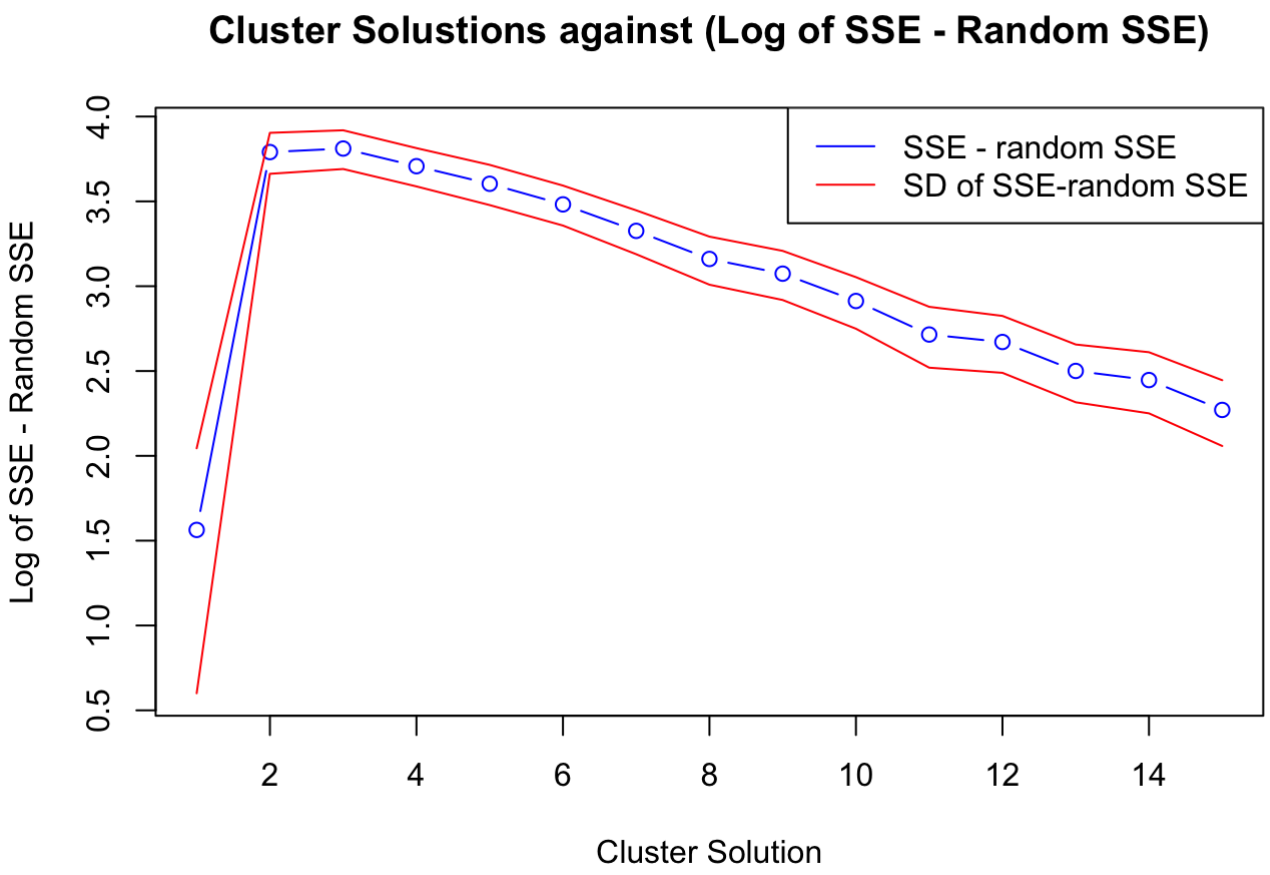
```
}

# Plot within groups SSE against all tested cluster solutions for actual and randomized
 data - 1st: Log scale, 2nd: Normal scale

xrange <- range(1:n.lev)
yrange <- range(log(rand.mat),log(wss))
plot(xrange,yrange, type='n', xlab='Cluster Solution', ylab='Log of Within Group SSE', m
ain='Cluster Solutions against Log of SSE')
for (i in 1:250) lines(log(rand.mat[,i]),type='l',col='red')
lines(log(wss), type="b", col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)
```




```
yrange <- range(rand.mat,wss)
plot(xrange,yrange, type='n', xlab="Cluster Solution", ylab="Within Groups SSE", main="C
luster Solutions against SSE")
for (i in 1:250) lines(rand.mat[,i],type='l',col='red')
lines(1:n.lev, wss, type="b", col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)
```




```
# Calculate the mean and standard deviation of difference between SSE of actual data and
SSE of 250 randomized datasets
r.sse <- matrix(0,dim(rand.mat)[1],dim(rand.mat)[2])
wss.1 <- as.matrix(wss)
for (i in 1:dim(r.sse)[2]) {
  r.temp <- abs(rand.mat[,i]-wss.1[,1])
  r.sse[,i] <- r.temp}
r.sse.m <- apply(r.sse,1,mean)
r.sse.sd <- apply(r.sse,1,sd)
r.sse.plus <- r.sse.m + r.sse.sd
r.sse.min <- r.sse.m - r.sse.sd

# Plot differeince between actual SSE mean SSE from 250 randomized datasets - 1st: Log s
cale, 2nd: Normal scale

xrange <- range(1:n.lev)
yrange <- range(log(r.sse.plus),log(r.sse.min))
plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='Log of SSE - Random SSE', ma
in='Cluster Solustions against (Log of SSE - Random SSE)')
lines(log(r.sse.m), type="b", col='blue')
lines(log(r.sse.plus), type='l', col='red')
lines(log(r.sse.min), type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), l
ty=1)
```



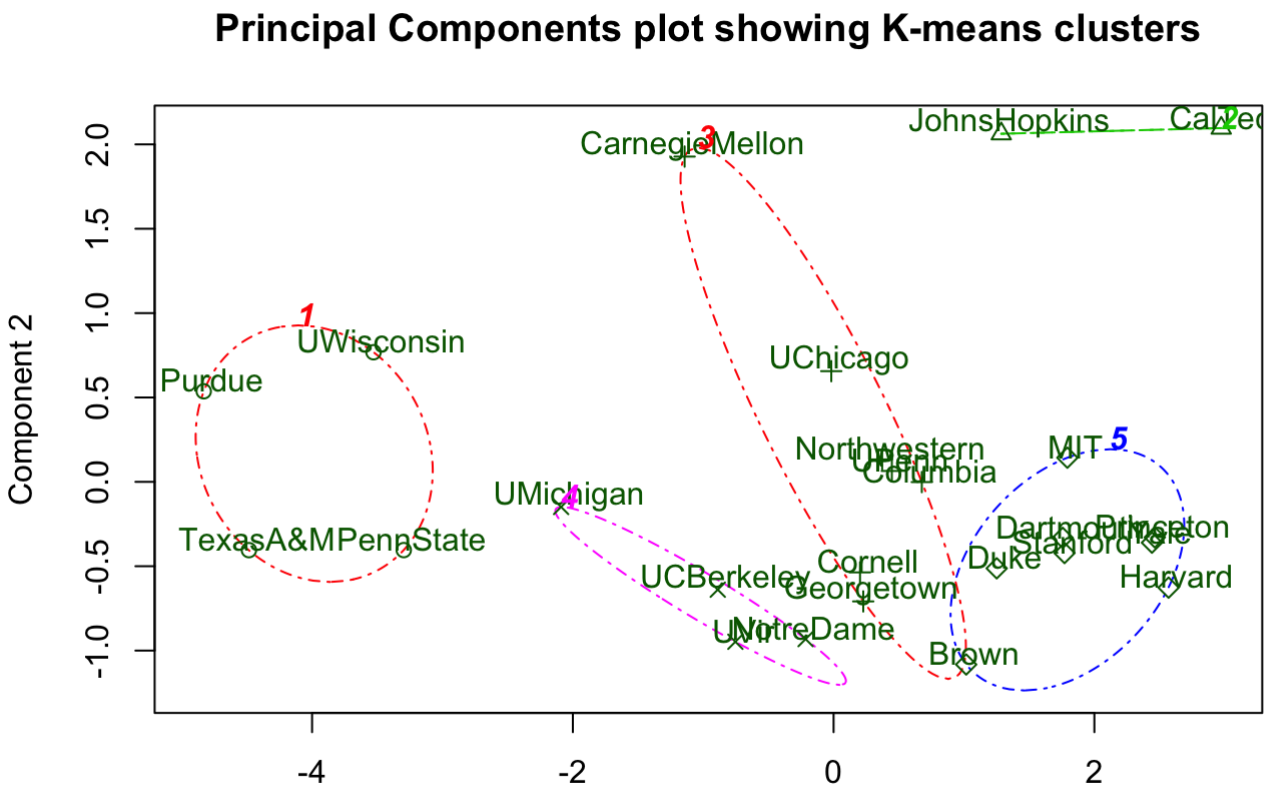

```
xrange <- range(1:n.lev)
yrange <- range(r.sse.plus,r.sse.min)
plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='SSE - Random SSE', main='Clu
ster Solutions against (SSE - Random SSE)')
lines(r.sse.m, type="b", col='blue')
lines(r.sse.plus, type='l', col='red')
lines(r.sse.min, type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), l
ty=1)
```


```
# Ask for user input - Select the appropriate number of clusters
#choose.clust <- function(){readline("What clustering solution would you like to use?
 ")}
#clust.level <- as.integer(choose.clust())
clust.level <- 5

# Apply K-means cluster solutions - append clusters to CSV file
fit <- kmeans(kdata, clust.level)
aggregate(kdata, by=list(fit$cluster), FUN=mean)
```

Group.1 <int>	SAT <dbl>	Top10 <dbl>	Accept <dbl>	SFRatio <dbl>	Expenses <dbl>	Grad <dbl>
1	-1.84224292	-1.8067438	1.4682954574	1.5285218	-1.4007248	-1.62064082
2	0.89783590	0.6213602	-0.1699892315	-1.7720453	2.0213426	-0.35675378
3	0.06159899	-0.1246205	-0.0003243245	-0.2342202	0.1152121	-0.01067158
4	-0.42997188	0.0866552	0.5494742845	0.6203686	-0.8929871	0.03548702
5	0.85774931	0.8137472	-0.9661037791	-0.4264912	0.5407097	0.89110298

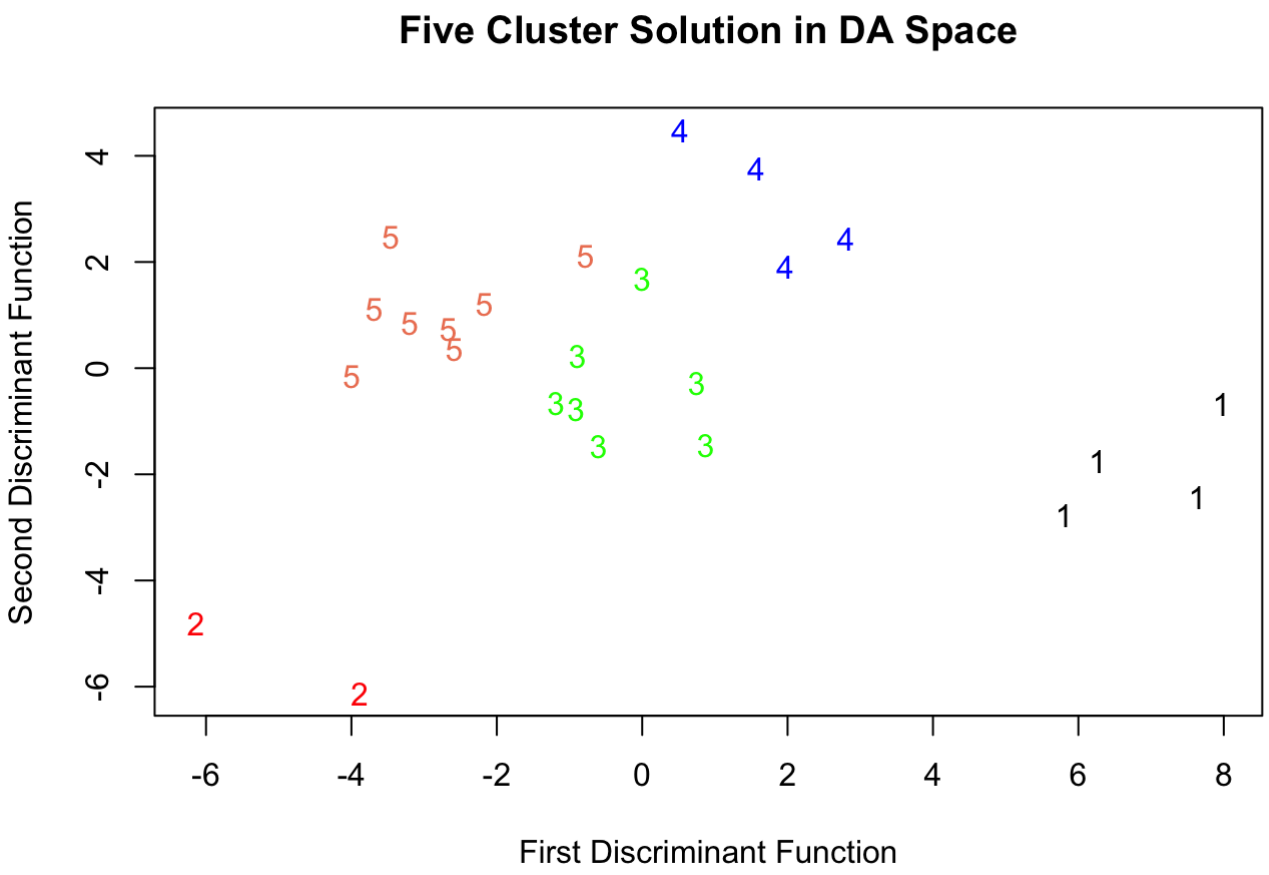
5 rows

```
clust.out <- fit$cluster
kclust <- as.matrix(clust.out)
kclust.out <- cbind(kclust, kdata)
write.table(kclust.out, file="kmeans_out.csv", sep=",")

# Display Principal Components plot of data with clusters identified

clusplot(kdata, fit$cluster, shade=F, labels=2, lines=0, color=T, lty=4, main='Principal
Components plot showing K-means clusters')
```



These two components explain 90.19 % of the point variability.

```
#Make plot of five cluster solution in space desginated by first two
#  two discriminant functions

plotcluster(kdata, fit$cluster, main="Five Cluster Solution in DA Space",
            xlab="First Discriminant Function", ylab="Second Discriminant Function")
```




```
# end of script
```