multipset5

Daniel Kim

3/31/2020

```
college <- read.csv("~/Downloads/Universities.csv")
college
```

```
##        University   SAT Top10 Accept SFRatio Expenses Grad
## 1         Harvard 14.00    91     14      11   39.525   97
## 2       Princeton 13.75    91     14       8   30.220   95
## 3            Yale 13.75    95     19      11   43.514   96
## 4        Stanford 13.60    90     20      12   36.450   93
## 5             MIT 13.80    94     30      10   34.870   91
## 6            Duke 13.15    90     30      12   31.585   95
## 7         CalTech 14.15   100     25       6   63.575   81
## 8       Dartmouth 13.40    89     23      10   32.162   95
## 9           Brown 13.10    89     22      13   22.704   94
## 10   JohnsHopkins 13.05    75     44       7   58.691   87
## 11       UChicago 12.90    75     50      13   38.380   87
## 12          UPenn 12.85    80     36      11   27.553   90
## 13        Cornell 12.80    83     33      13   21.864   90
## 14   Northwestern 12.60    85     39      11   28.052   89
## 15       Columbia 13.10    76     24      12   31.510   88
## 16      NotreDame 12.55    81     42      13   15.122   94
## 17           UVir 12.25    77     44      14   13.349   92
## 18     Georgetown 12.55    74     24      12   20.126   92
## 19 CarnegieMellon 12.60    62     59       9   25.026   72
## 20      UMichigan 11.80    65     68      16   15.470   85
## 21     UCBerkeley 12.40    95     40      17   15.140   78
## 22     UWisconsin 10.85    40     69      15   11.857   71
## 23      PennState 10.81    38     54      18   10.185   80
## 24         Purdue 10.05    28     90      19    9.066   69
## 25       TexasA&M 10.75    49     67      25    8.704   67
```

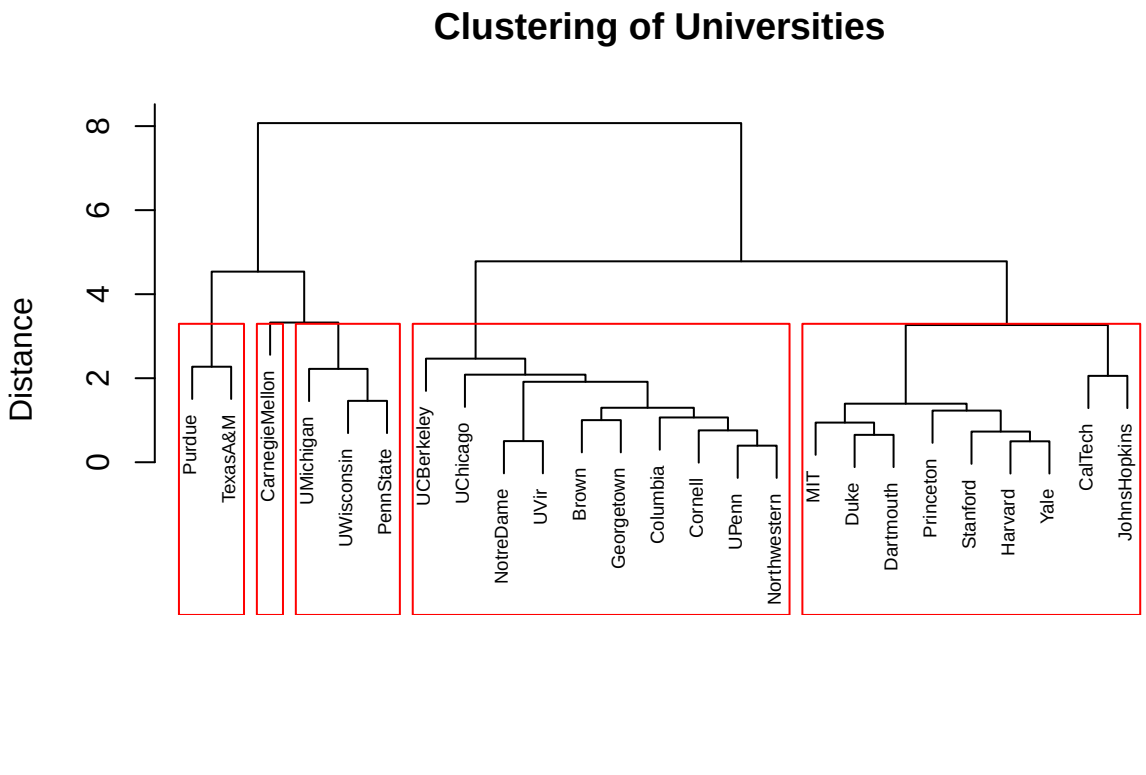
1) The variables seem to be mainly continuous variable so I think Euclidean distance as a metric would work in order to measure distance between points. The standard deviations of the variables seem to vary so I will scale/standardize my data in part 2.

```
var(college[,c(2, 3, 4, 5, 6, 7)])
```

```
##                SAT      Top10     Accept    SFRatio   Expenses        Grad
## SAT       1.174184   19.42697  -18.93633  -3.581217   12.17599    7.338783
## Top10    19.426967  377.67667 -329.39167 -50.860000  171.41354  131.306667
## Accept  -18.936333 -329.39167  389.16667  50.683333 -158.91183 -146.441667
## SFRatio  -3.581217  -50.86000   50.68333  16.543333  -45.87133  -20.665000
## Expenses 12.175995  171.41354 -158.91183 -45.871333  208.07725   51.425625
## Grad      7.338783  131.30667 -146.44167 -20.665000   51.42562   82.043333
```

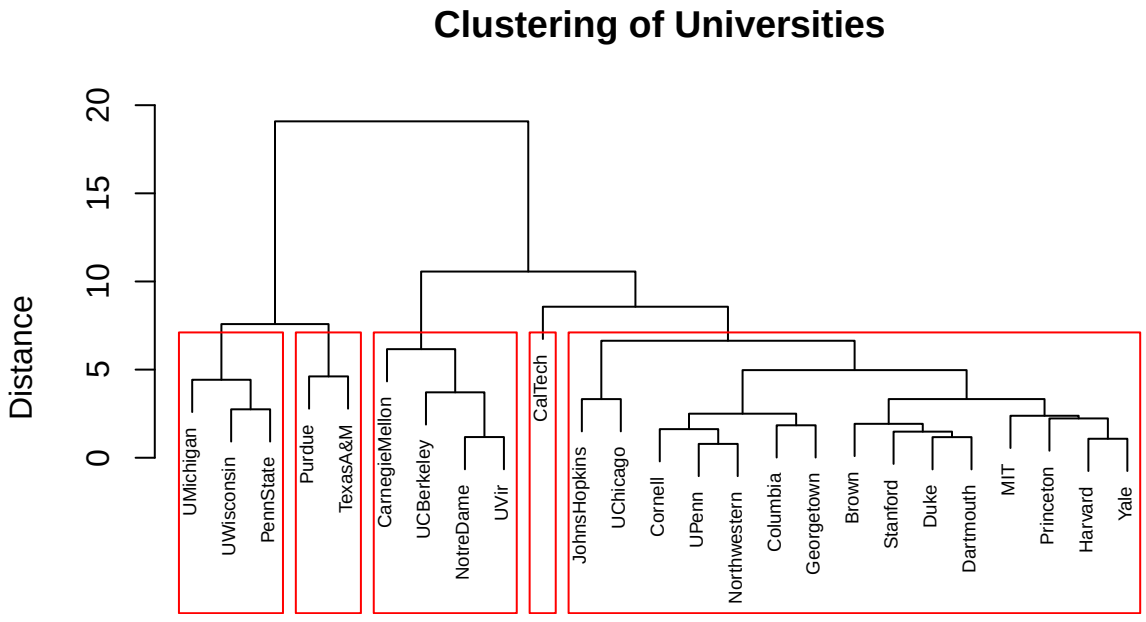
2)

```
collegenorm <- college[,c("SAT","Top10","Accept","SFRatio","Expenses","Grad")]
rownames(collegenorm) <- college[,1]
collegenorm <- scale(na.omit(collegenorm)) # scaling my variables

#get the distance matrix
dist1 <- dist(collegenorm, method="euclidean")


clust1 <- hclust(dist1)

#draw the dendrogram
plot(clust1,labels= rownames(collegenorm), cex=0.6, xlab="",ylab="Distance",main="Clustering of Universi
rect.hclust(clust1, k =5)
```



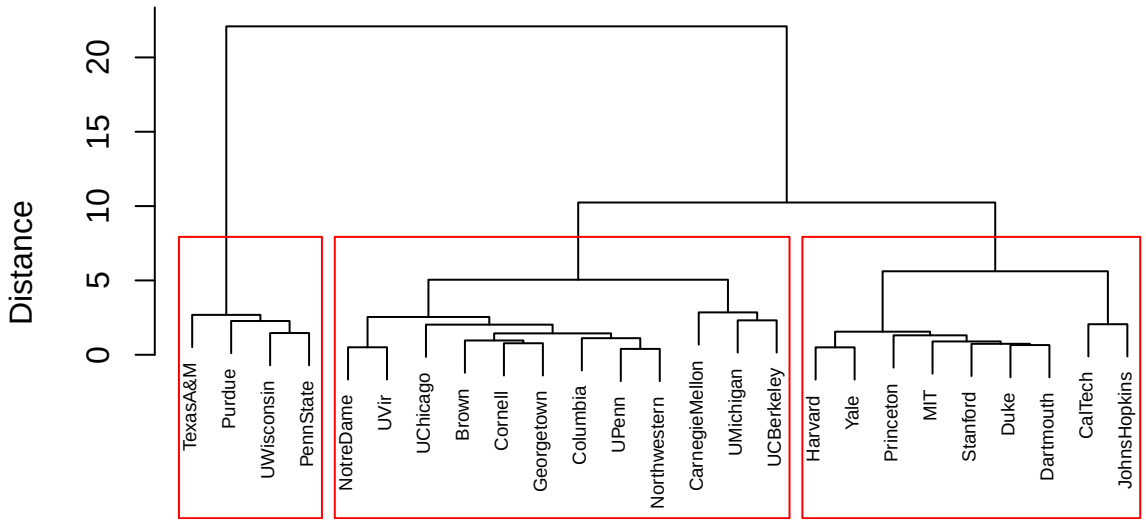
Using euclidean distance metrics and complete linkage method, there appears to be perhaps 5 main groups asssociated with the clustering of universities.

```
dist2 <- dist(collegenorm, method="manhattan")
clust2 <- hclust(dist2)
plot(clust2,labels= rownames(collegenorm), cex=0.6, xlab="",ylab="Distance",main="Clustering of Universi
rect.hclust(clust2, k = 5)
```



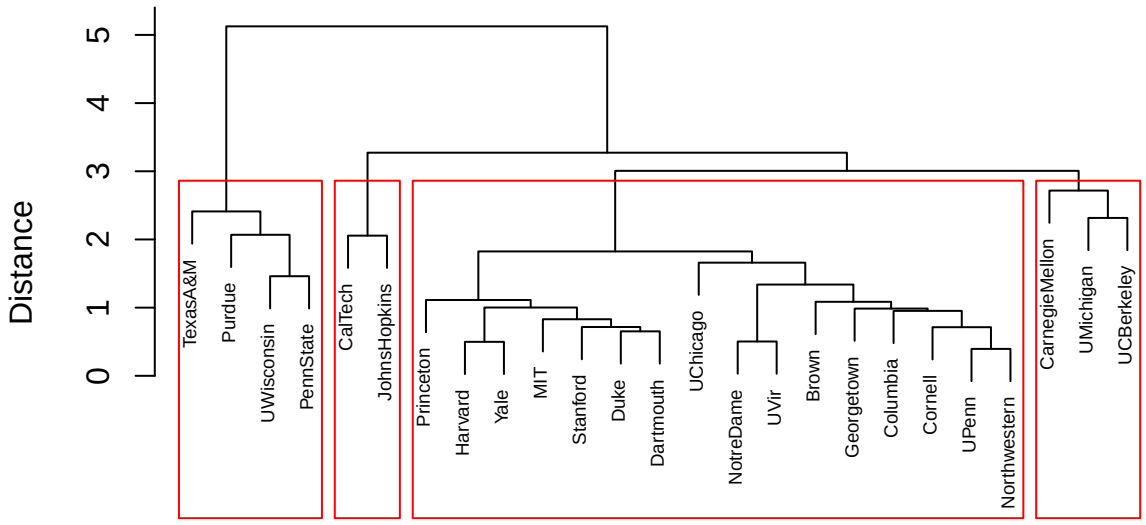
Using manhattan distance metrics and complete linkage method, there seems to be a bunch larger abudnace of smaller subgroups, mot likely attributted to the non-euclidean distance metric

```
dist3 <- dist(collegenorm, method="euclidean")
clust3 <- hclust(dist3, method = "ward.D")
plot(clust3,labels= rownames(collegenorm), cex=0.6, xlab="",ylab="Distance",main="Clustering of Univers
rect.hclust(clust3, k = 3)
```



Using euclidean distance metrics and ward linkage method, you could argue for maybe 3 main groups, attributted to the ward method of minimizing sum of squares.

```
dist4 <- dist(collegenorm, method="euclidean")
clust4 <- hclust(dist4, method = "average")
plot(clust4,labels= rownames(collegenorm), cex=0.6, xlab="",ylab="Distance",main="Clustering of Univers
rect.hclust(clust4, k = 4)
```

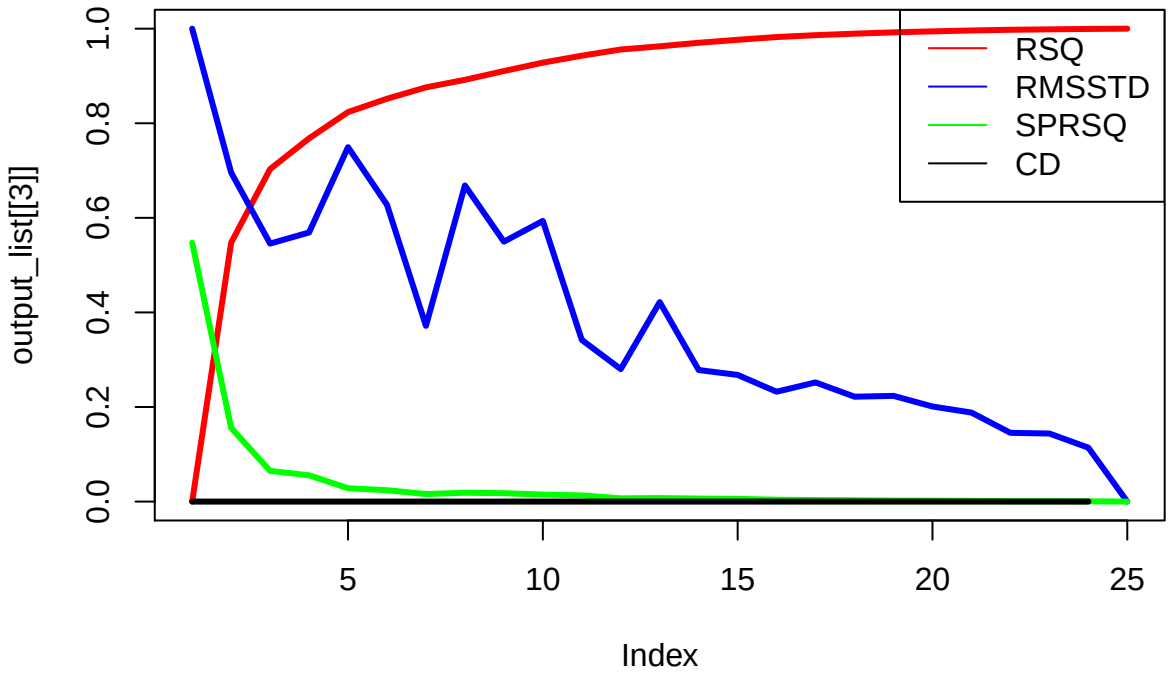


Using euclidean distance metrics and average linkage method, you could argue for around 4 cluster groups. This method is a space conserving method which could be the reason why.

3)

```
source("http://reuningscherer.net/stat660/R/HClusEval.R.txt")
hclus_eval(collegenorm, dist_m = 'euclidean', clus_m = 'ward', plot_op = T)
```

```
## [1] "Creating Distance Matrix using euclidean"
## [1] "Clustering using ward"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Clustering Complete. Access the Cluster object in first element of output"
## [1] "Calculating RMSSTD"
## [1] "RMSSTD Done. Access in Element 2"
## [1] "Calculating RSQ"
## [1] "RSQ Done. Access in Element 3"
## [1] "Calculating SPRSQ"
## [1] "SPRSQ Done. Access in Element 4"
## [1] "Calculating Cluster Dist. "
## [1] "CD Done. Access in Element 5"
```




```
## [[1]]
##
## Call:
## hclust(d = dist1, method = clus_m)
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 25
##
##
## [[2]]
##  [1] 1.0000000 0.6957217 0.5455504 0.5690948 0.7494865 0.6275674 0.3716089
##  [8] 0.6684786 0.5496442 0.5934966 0.3418013 0.2801410 0.4217875 0.2781637
## [15] 0.2678412 0.2322383 0.2519726 0.2217753 0.2235236 0.2010975 0.1882320
## [22] 0.1453734 0.1439020 0.1139935 0.0000000
##
## [[3]]
##  [1] 0.0000000 0.5474127 0.7031216 0.7680337 0.8236317 0.8518232 0.8758776
##  [8] 0.8918210 0.9104404 0.9282034 0.9428800 0.9559672 0.9627676 0.9701803
## [15] 0.9765147 0.9823232 0.9862763 0.9894854 0.9922634 0.9943452 0.9962389
## [22] 0.9977152 0.9985957 0.9994586 1.0000000
##
## [[4]]
##  [1] 0.5474127146 0.1557088738 0.0649120744 0.0555979982 0.0281915177
##  [6] 0.0240543736 0.0159434824 0.0186193184 0.0177630311 0.0146765903
## [11] 0.0130872221 0.0068004396 0.0074126959 0.0063344024 0.0058084457
## [16] 0.0039531123 0.0032090666 0.0027780177 0.0020817835 0.0018937129
## [21] 0.0014763040 0.0008805596 0.0008628244 0.0005414387 0.0000000000
##
## [[5]]
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

It looks like there are around 5 cluster groups. While the RMSSTD and CD lines offer little information, the

points where the RSQ and SPRSQ curves start to level out is around 5 cluster groups.

4)

```
km1 <- kmeans(collegenorm,centers=5)
km1
```

```
## K-means clustering with 5 clusters of sizes 4, 2, 7, 9, 3
##
## Cluster means:
##           SAT      Top10     Accept     SFRatio   Expenses       Grad
## 1 -1.89129229 -1.9414523  1.5612876  1.60546806 -1.2086753 -1.6527233
## 2  0.86342006  0.5670502 -0.2382484 -1.52925136  2.3393604 -0.3002944
## 3  0.89637905  0.7692006 -0.9008542 -0.52824852  0.5606384  0.8668162
## 4  0.07386915  0.1811267 -0.2185352 -0.06774818 -0.2143826  0.4357213
## 5 -0.36704889 -0.1276120  0.8347143  0.31470124 -0.6130148 -0.9259078
##
## Clustering vector:
##        Harvard      Princeton           Yale       Stanford            MIT
##              3              3              3              3              3
##           Duke        CalTech      Dartmouth          Brown   JohnsHopkins
##              3              2              3              4              2
##       UChicago          UPenn        Cornell   Northwestern       Columbia
##              4              4              4              4              4
##      NotreDame           UVir     Georgetown CarnegieMellon      UMichigan
##              4              4              4              5              5
##     UCBerkeley     UWisconsin      PennState         Purdue       TexasA&M
##              5              1              1              1              1
##
## Within cluster sum of squares by cluster:
## [1] 7.089134 2.113429 2.825243 6.628474 6.740760
##  (between_SS / total_SS =  82.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
for (i in 1:5){
  print(paste("Universities in Cluster ",i))
  print(college$University[km1$cluster==i])
  print (" ")
}
```

```
## [1] "Universities in Cluster  1"
## [1] UWisconsin PennState  Purdue     TexasA&M
## 25 Levels: Brown CalTech CarnegieMellon Columbia Cornell Dartmouth ... Yale
## [1] " "
## [1] "Universities in Cluster  2"
## [1] CalTech      JohnsHopkins
## 25 Levels: Brown CalTech CarnegieMellon Columbia Cornell Dartmouth ... Yale
## [1] " "
## [1] "Universities in Cluster  3"
## [1] Harvard   Princeton Yale      Stanford  MIT       Duke      Dartmouth
## 25 Levels: Brown CalTech CarnegieMellon Columbia Cornell Dartmouth ... Yale
## [1] " "
```

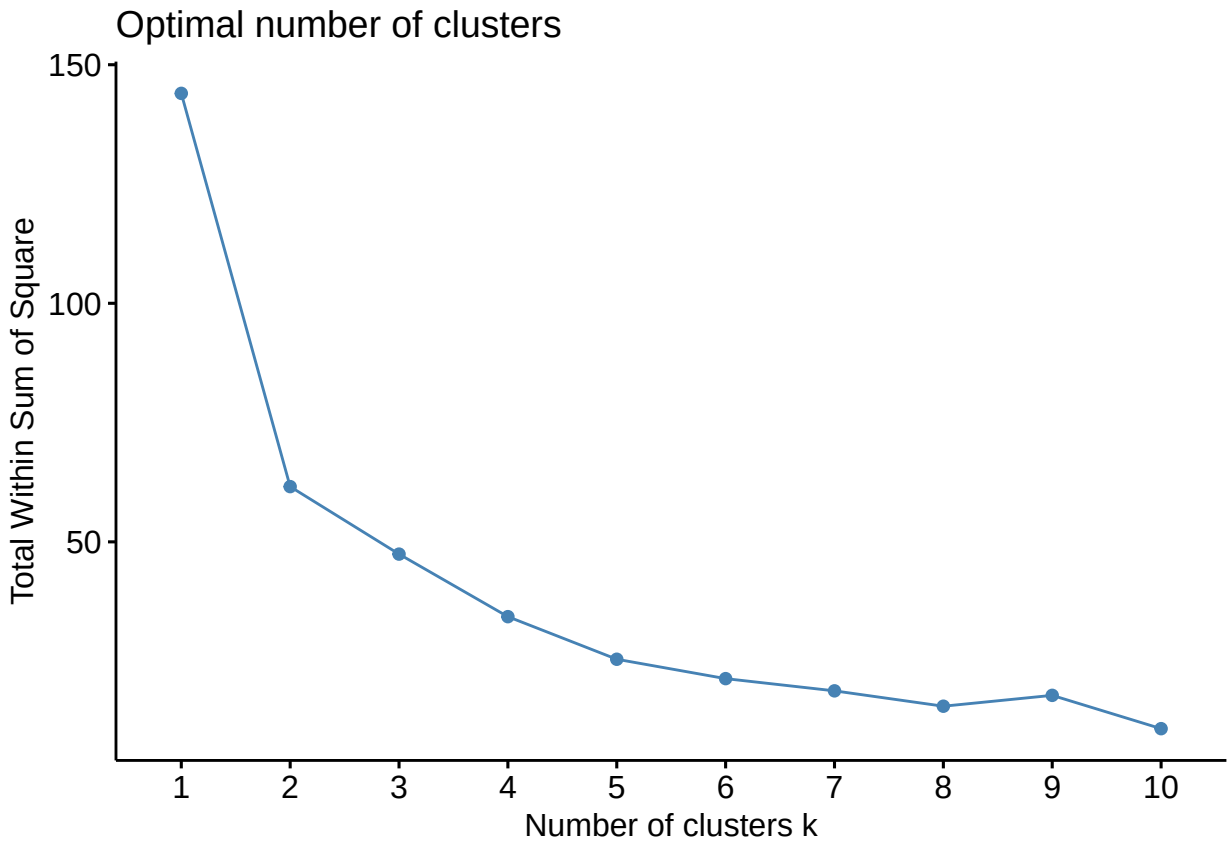
```
## [1] "Universities in Cluster  4"
## [1] Brown        UChicago     UPenn        Cornell      Northwestern
## [6] Columbia     NotreDame    UVir         Georgetown
## 25 Levels: Brown CalTech CarnegieMellon Columbia Cornell Dartmouth ... Yale
## [1] " "
## [1] "Universities in Cluster  5"
## [1] CarnegieMellon UMichigan      UCBerkeley
## 25 Levels: Brown CalTech CarnegieMellon Columbia Cornell Dartmouth ... Yale
## [1] " "
```

```
set.seed(123)
library(factoextra)
```

```
## Loading required package: ggplot2
```

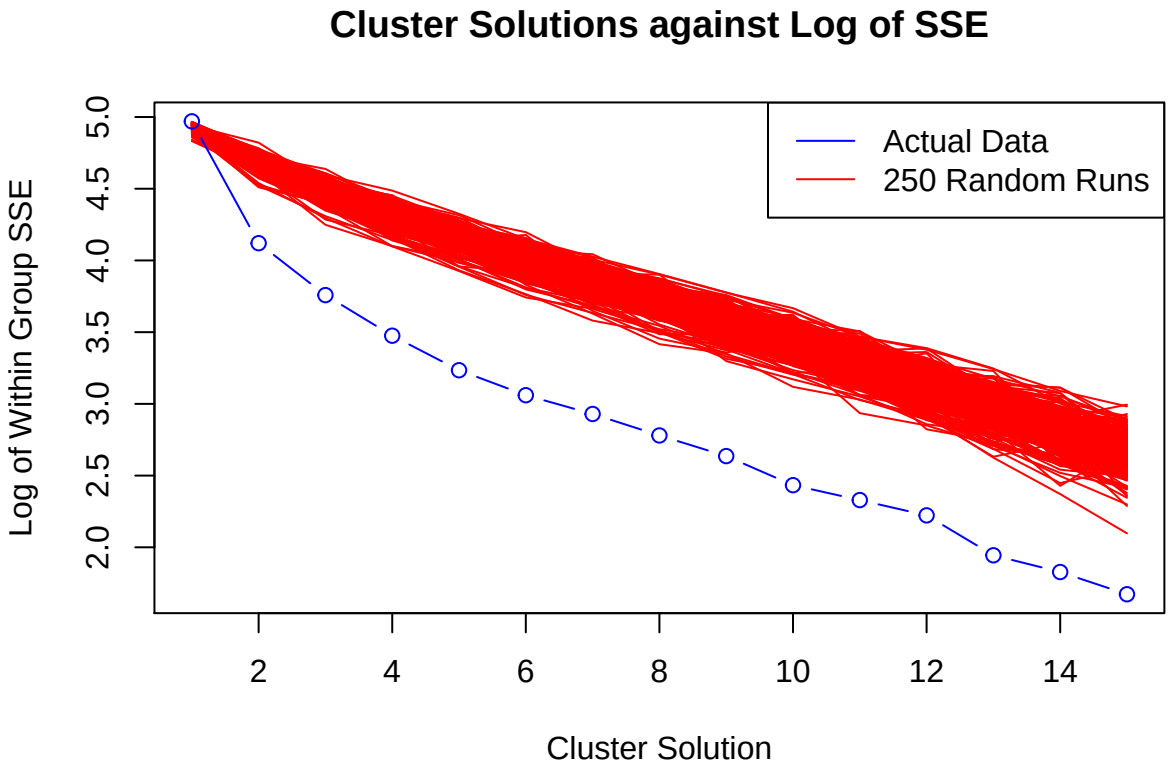
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_nbclust(collegenorm, kmeans, method = "wss")
```




```
kdata <- collegenorm
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 solutions selected b
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 input 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

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', main='Cluster Sol
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)
```



There seems to be around 5 groups when looking at the k-means result. Looking at the SSE plotted against the cluster groups for the actual data against 250 random runs, the point where the distance between the two stops cahnging is around 5 groups.

5)

Based on the variety of dendrograms, we would reason that there should be somewhere around 4-6 groups among our data when clustering. The R square and semi-partial R squared graphs seem to place the number of groups around 5 and this is supplanted by the k-means data where we can graph the sum of squares within groups against cluster groups to see that the number of groups present seems to be around 5.