cluster

Table of Contents

[AGGLOMERATIVE HIERARCHICAL CLUSTERING 2](file:///C:\Users\kgund\Dropbox\2019%20CS%20450\Week_10\cluster.docx#_Toc3627823)

[USING K-MEANS 5](file:///C:\Users\kgund\Dropbox\2019%20CS%20450\Week_10\cluster.docx#_Toc3627824)

library(datasets)  
library(cluster)

## Warning: package 'cluster' was built under R version 3.5.3

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages ------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0 v readr 1.1.1  
## v tibble 1.4.2 v purrr 0.2.5  
## v tidyr 0.8.1 v stringr 1.3.1  
## v ggplot2 3.0.0 v forcats 0.3.0

## -- Conflicts ---------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(RColorBrewer)

## Warning: package 'RColorBrewer' was built under R version 3.5.2

library(animation)

## Warning: package 'animation' was built under R version 3.5.3

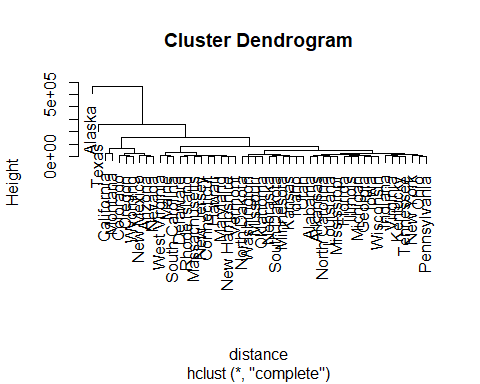
# AGGLOMERATIVE HIERARCHICAL CLUSTERING

Step 1

##Step 1   
data = state.x77  
data <- data.frame(data)

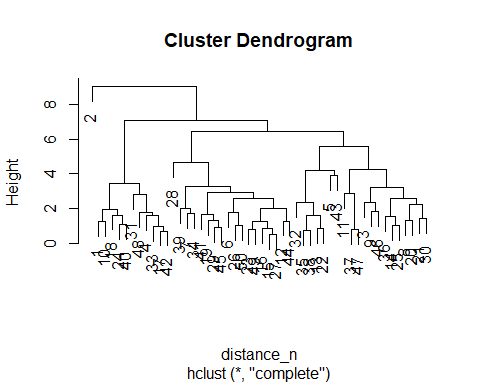
Step 2

##Step 2  
#Dendrogram of non-normalized data  
# first compute a distance matrix  
distance = dist(as.matrix(data))  
# now perform the clustering  
hc = hclust(distance)  
# finally, plot the dendrogram  
plot(hc)



Step 3

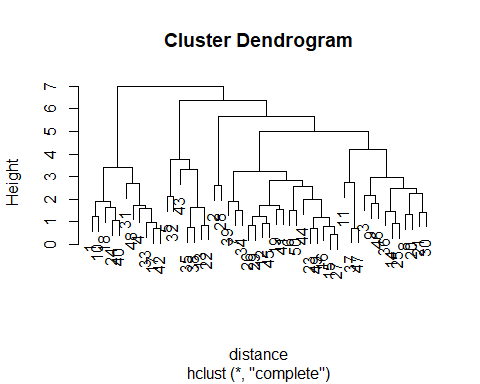
##Step 3  
#Dendrogram of normalized data  
#summary(data)  
#Scaling the data  
data\_scaled <- data %>%  
 mutate(pop\_scal = scale(Population),  
 income\_scal = scale(Income),  
 illit\_scal = scale(Illiteracy),  
 life\_scal = scale(Life.Exp),  
 murder\_scal = scale(Murder),  
 grad\_scal = scale(HS.Grad),  
 frost\_scal = scale(Frost),  
 area\_scal = scale(Area)) %>%  
 select(-c(Population, Income, Illiteracy, Life.Exp,   
 Murder, HS.Grad, Frost, Area))  
  
# distance matrix  
distance\_n = dist(as.matrix(data\_scaled))  
# perform clustering  
hc\_n = hclust(distance\_n)  
# plot the dendrogram  
plot(hc\_n)



When I scaled the data it took away the name of the states in alphabetical order and assigned them a number that corresponds with its place alphabetically.When the data was normalized the clustering changed. The normalized data seemed to do a better job of clustering into more distinct groups. It seems that the attributes impact the clustering more when its normalized.

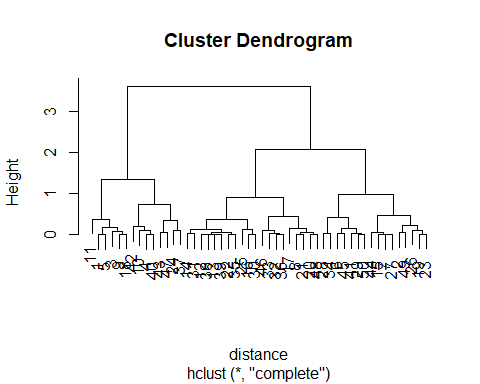
Step 4

##Step 4  
data\_scaled\_a <- data\_scaled %>%   
 select(-area\_scal)  
# compute a distance matrix  
distance = dist(as.matrix(data\_scaled\_a))  
# perform the clustering  
hc\_a = hclust(distance)  
# plot the dendrogram  
plot(hc\_a)

 When area was removed, the clusters seemed to be more balanced across the dendrogram. It makes sense that when you remove the area of each state it everything seems more balanced, this is because not each state has the same area.

Step 5

##Step 5  
data\_scaled\_f <- data\_scaled[,7]  
# compute a distance matrix  
distance = dist(as.matrix(data\_scaled\_f))  
# perform the clustering  
hc\_f = hclust(distance)  
# plot the dendrogram  
plot(hc\_f)

 States with similar weather patterns are clustered together.

# USING K-MEANS

Step 1

##Step 1.  
#use nomralized data

Step 2

##Step 2.  
# Cluster into k=3 clusters:  
pc\_cluster = kmeans(data\_scaled, 3)  
summary(pc\_cluster)

## Length Class Mode   
## cluster 50 -none- numeric  
## centers 24 -none- numeric  
## totss 1 -none- numeric  
## withinss 3 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 3 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

print(pc\_cluster$cluster)

## [1] 1 2 1 1 3 3 3 3 1 1 3 3 3 3 3 3 1 1 3 3 3 3 3 1 3 3 3 3 3 3 1 1 1 3 3  
## [36] 3 3 3 3 1 3 1 1 3 3 1 3 1 3 3

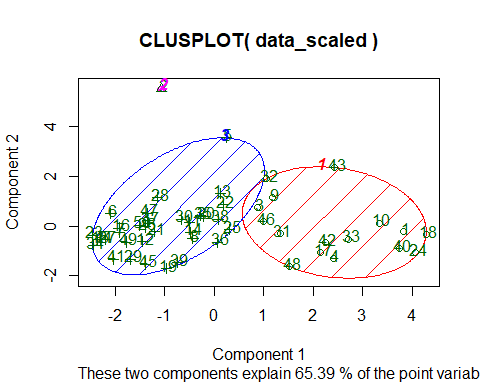
print(pc\_cluster$centers)

## pop\_scal income\_scal illit\_scal life\_scal murder\_scal grad\_scal  
## 1 0.18763727 -0.7973132 1.1873842 -0.8863645 0.9913208 -1.0270524  
## 2 -0.86939802 3.0582456 0.5413980 -1.1685098 1.0624293 1.6828035  
## 3 -0.06463025 0.2939020 -0.5921074 0.4651619 -0.5128352 0.4469708  
## frost\_scal area\_scal  
## 1 -0.8493032 -0.04108231  
## 2 0.9145676 5.80934967  
## 3 0.3840692 -0.15612220

print(pc\_cluster$size)

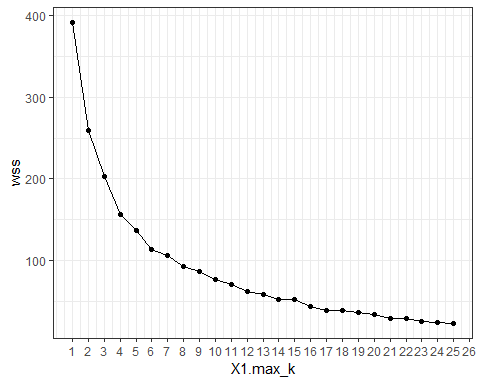
## [1] 16 1 33

clusplot(data\_scaled, pc\_cluster$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)

 I would think they are grouped this way because population plays a big factor in their groupings.

Step 3

#Step 3.  
# Cluster into k=3 clusters:  
pc\_cluster = kmeans(data\_scaled, 3)  
# compute the total within clusters sum of squares  
kmean\_withinss <- function(k) {  
 cluster <- kmeans(data\_scaled, k)  
 return (cluster$tot.withinss)  
}  
# Set maximum cluster   
max\_k <-25   
# Run algorithm over a range of k   
wss <- sapply(1:max\_k, kmean\_withinss)  
elbow <-data.frame(1:max\_k, wss)  
ggplot(elbow, aes(x = X1.max\_k, y = wss)) +  
 geom\_point() +  
 geom\_line() +  
 scale\_x\_continuous(breaks = seq(1, 26, by = 1)) +  
 theme\_bw()



Step 4

#Step 4.  
#Best looks like 7  
pc\_cluster\_2 <- kmeans(data\_scaled, 7)

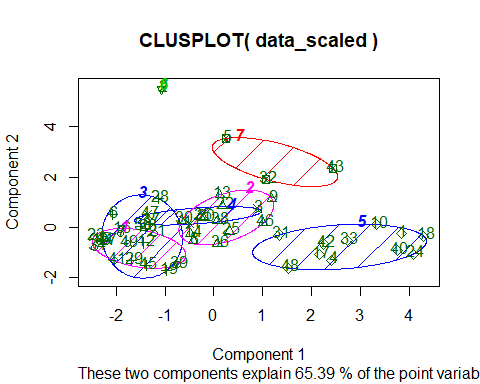
Step 5

##Step 5  
data\_scale = scale(data)  
pc\_cluster\_3 <- kmeans(data\_scale, 7)  
  
print(pc\_cluster\_3$cluster)

## Alabama Alaska Arizona Arkansas California   
## 7 4 5 2 6   
## Colorado Connecticut Delaware Florida Georgia   
## 3 1 5 6 7   
## Hawaii Idaho Illinois Indiana Iowa   
## 1 3 6 5 1   
## Kansas Kentucky Louisiana Maine Maryland   
## 1 2 7 3 5   
## Massachusetts Michigan Minnesota Mississippi Missouri   
## 1 6 1 7 5   
## Montana Nebraska Nevada New Hampshire New Jersey   
## 3 1 3 3 5   
## New Mexico New York North Carolina North Dakota Ohio   
## 2 6 2 1 6   
## Oklahoma Oregon Pennsylvania Rhode Island South Carolina   
## 5 1 6 1 7   
## South Dakota Tennessee Texas Utah Vermont   
## 3 2 6 3 3   
## Virginia Washington West Virginia Wisconsin Wyoming   
## 5 1 2 1 3

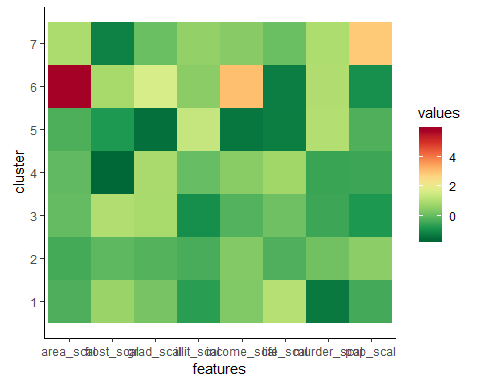
Step 6

##Step 6  
clusplot(data\_scaled, pc\_cluster\_2$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)



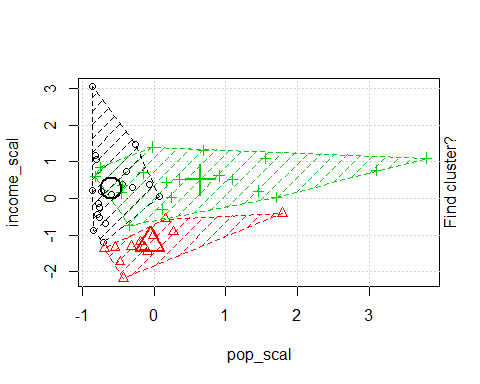
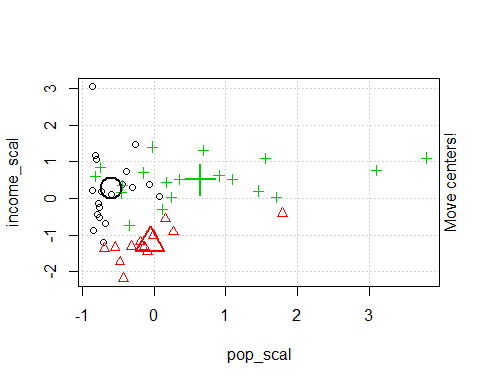
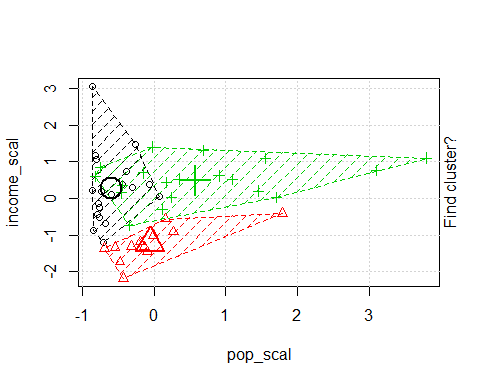
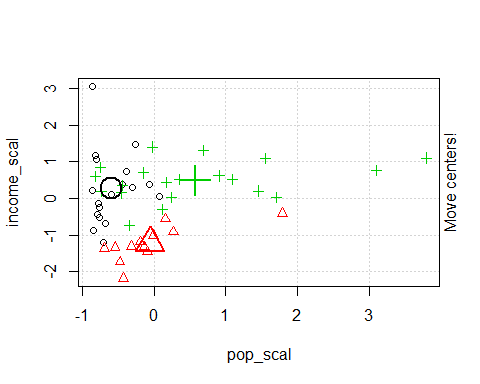
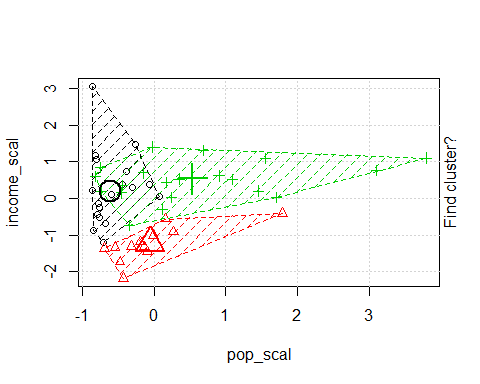
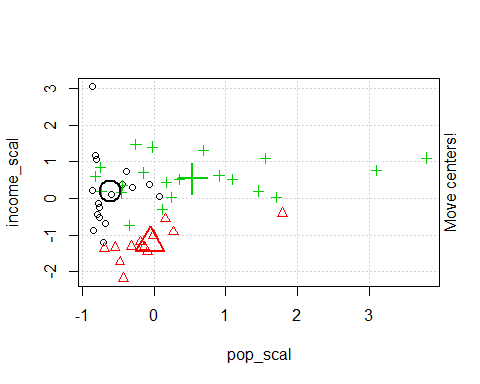
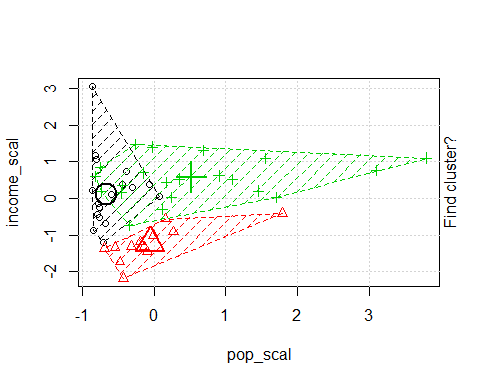
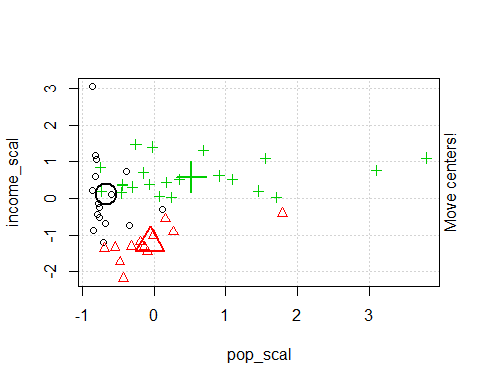
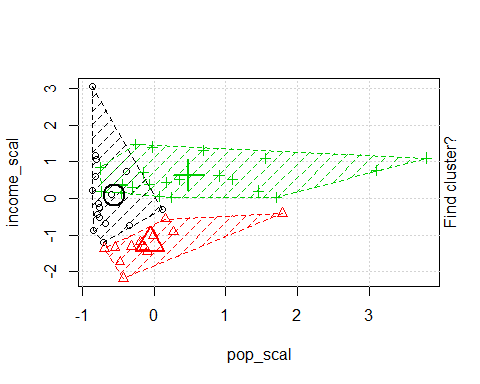
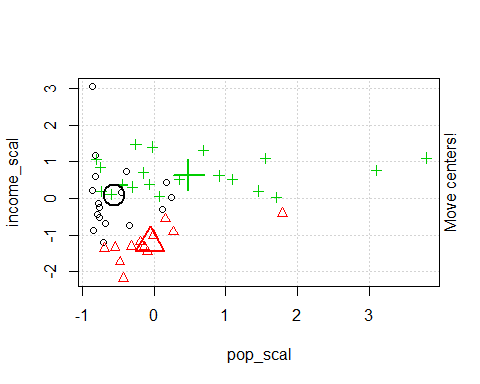
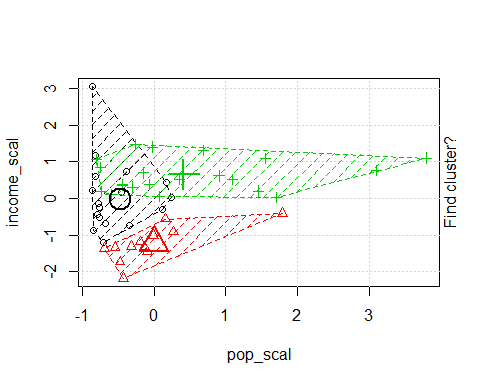
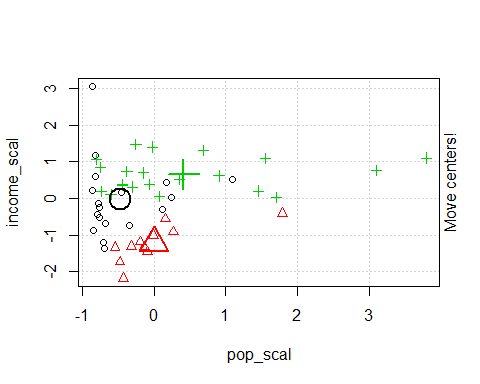
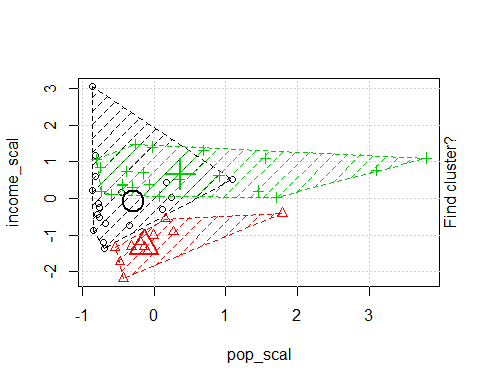
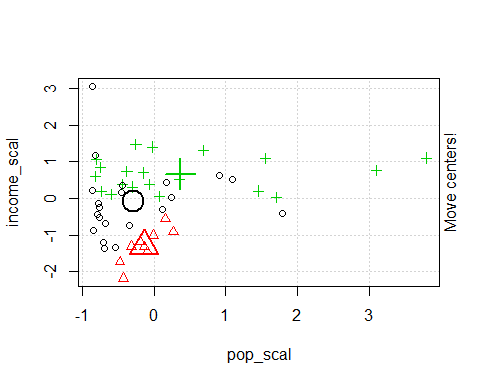
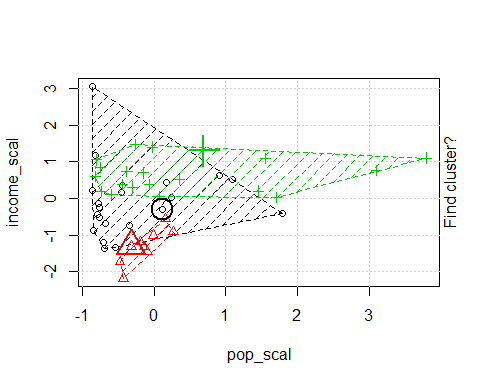
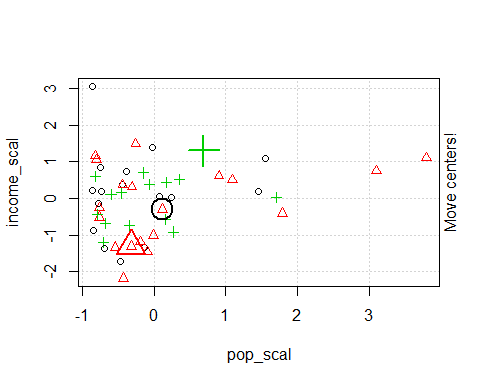
Step 7

##Step 7  
#get centers  
center <-pc\_cluster\_2$centers  
# create dataset with the cluster number  
cluster <- c(1:7)  
center\_df <- data.frame(cluster, center)  
# Reshape the data  
center\_reshape <- gather(center\_df, features, values, pop\_scal: area\_scal)  
#Colors for plotting   
# Create the palette  
hm.palette <-colorRampPalette(rev(brewer.pal(10, 'RdYlGn')),space='Lab')  
#Visualize the clusters  
#see what the clusters look like.  
ggplot(data = center\_reshape, aes(x = features, y = cluster, fill = values)) +  
 scale\_y\_continuous(breaks = seq(1, 7, by = 1)) +  
 geom\_tile() +  
 coord\_equal() +  
 scale\_fill\_gradientn(colours = hm.palette(90)) +  
 theme\_classic()

 It is noticeable that population, income and area have the greatest variation among all of the attributes. If we know this I would say that they also have the greatest influence on the centers when running the clustering algorithm.

This is what the clustering algorithm looks like for every step on the data

#This is what the clustering algorithm looks like for every step on the data  
set.seed(2345)  
# par(mfrow=c(2,1))  
kmeans.ani(data\_scaled, 3)



# dev.off()

I choose Shows creativity and excels above and beyond requirements I put this because I spent a lot of time trying to get all of the steps completed and making graphs that would better help me understand how the algorithm worked. I feel that I went above because did a heat map of the clusters and did an animation of the cluster algorithm on the state data so you could see whats happening step by step.