Spring 2019

Predicting Cable Subscribers

Mid Semester Report

Mahbubul Hasan (mhasan1)

Elizabeth James (enjames)

Omer Khan (oakhan)

Jennifer Tichenor (jtchnor2)

MIS 7621 - 001

# Goal

Using information regarding an individual’s age, gender, income, home ownership, class segment, and number of children, we plan to predict whether or not the individual will subscribe to cable.

# Summary:

This data contains information on people who have subscribed to a certain kind of Cable TV service. The data contains just 300 observations and 7 variables.

The 7 variables are:   
**Age** - the age of the TV subscriber   
**Gender** - the gender of the TV subscriber   
**Income** - the income of the TV subscriber   
**kids** - the number of kids the TV subscriber has   
**ownHome** - if the TV subscriber owns the home or not   
**subscribe** - if they have subscribed to the TV services or not   
**segment** - the segment of the TV subscriber's subscription

# Preliminary Analysis of the Data

The data is based on 300 observations of people subscribed to cable TV. Cable subscribers range in age from 19 to 80. The median age is 39, while the average age is 41, so the ages skew just a little higher. Income has a minimum of $-5,183 and a maximum of $114,278. We may need to dig into why some incomes report as being negative to understand whether the data is valid or not. The median income is $52,014 while the average income is $50,937. The number of children range from 0 to 7 per subscriber, with the median being 1, while the mean is 1.27. A majority of subscribers have 1 child or no children at all. Additional observations include:

* Slightly more subscribers are women (52%).
* 87% of people polled have not subscribed to cable.
* 53% of people say they do not own their home.
* Given the options to categorize their lifestyle as suburb, urban, travelers, or transitioning,
  + 26% surveyed identified themselves as travelers,
  + 17% selected urban,
  + 23% chose transitioning or “moving up,” and
  + 33% selected suburb.

# Clustering

Clustering was conducted using the “stats” and “cluster” libraries, and all non-binary attributes were clustered against one another and displayed in a matrix plot, using k-means. K-values from three to seventeen were used for experimentation. Income to age best exemplifies the concept of clustering. The other pairings are relevant, but they are not as visually intriguing, as they are each primarily simply banded distributions and clusters.

Plot matrices for 3-, 5-, and 8- clusters are included as examples.

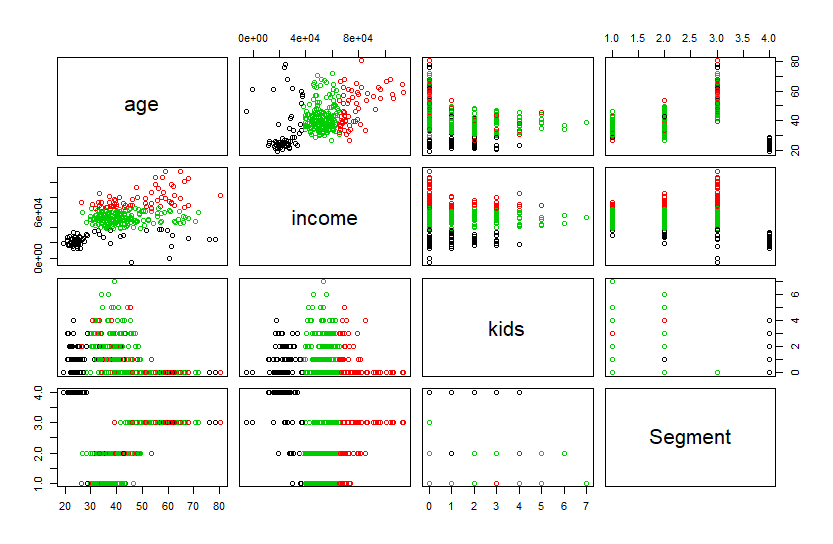


Figure 1: Three Clusters

Figure 3: Eight Clusters

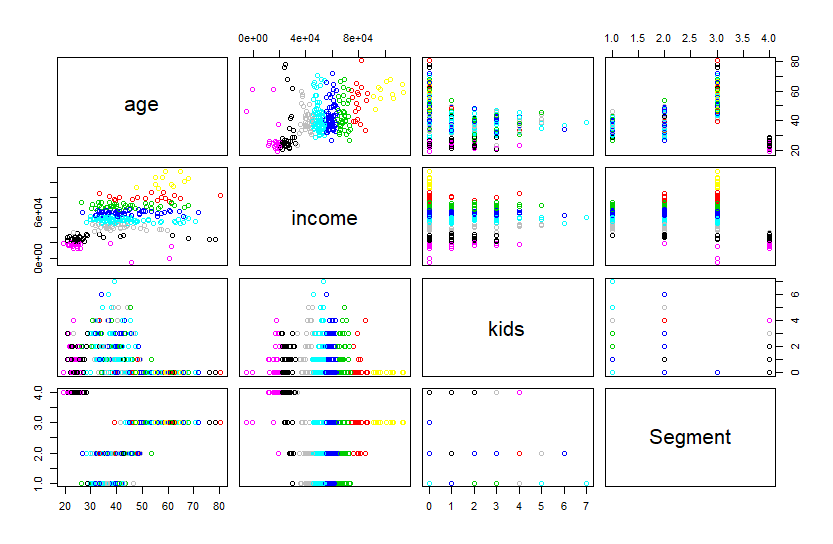
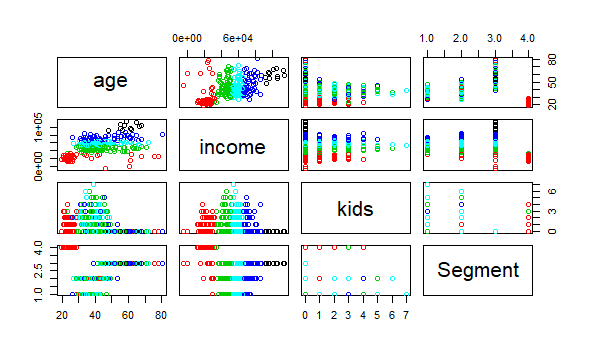


Figure 2: Five Clusters



## Elbow Method

The k-value for each cluster was then plotted against the total within sum of squares to determine the best k-value.

Using the graph in Figure 4, we see that “elbows” appear to occur at both k = 5 and k = 8.

## Silhouette

Figure 5 shows the Silhouette Method. The highest value clearly occurs at k = 3, indicating 3 clusters to be the optimal separation.

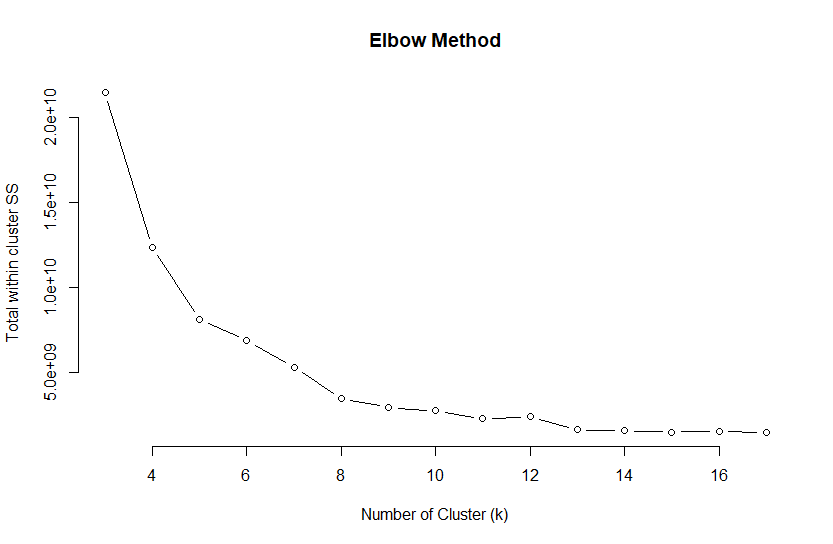


Figure 4: Elbow method

## Ratio

The ratio of intra- to inter-cluster distance was also calculated using the formula:

between sum of squares

total within sum of squares

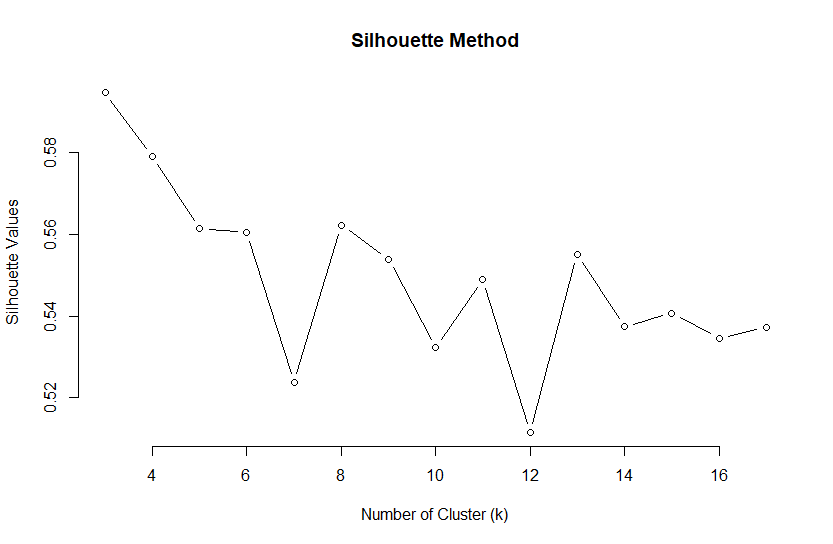


Figure 5: Silhouette Method

|  |  |
| --- | --- |
| K | Ratio |
| 3 | 4.638214 |
| 4 | 8.835721 |
| 5 | 14.01172 |
| 6 | 19.25559 |
| 7 | 24.18168 |
| 8 | 34.1379 |
| 9 | 29.48415 |
| 10 | 44.50326 |
| 11 | 58.52907 |
| 12 | 56.90828 |
| 13 | 61.89989 |
| 14 | 73.0981 |
| 15 | 134.4622 |
| 16 | 85.94473 |
| 17 | 136.1039 |

# Association Rule Mining

The data was transformed into a binary matrix, where 1 was assigned as a “yes” under a given attribute and 0 was assigned to a “no.” The matrix included the following attributes:

* kids0
* kids1
* kids2
* kids3
* kids4
* kids5
* kids6
* kids7
* group\_Male
* group\_Female
* group\_ownYes
* group\_ownNo
* group\_subYes
* group\_subNo
* group\_seg\_MovingUp
* group\_seg\_SubMix
* group\_seg\_Travelers
* group\_seg\_UrbanHip
* ageUnder25
* age25to35
* age35to45
* age45to55
* age55to65
* ageOver65
* incomeUnder10K
* income10to40K
* income40to70K
* income70to100K
* incomeOver100K

Most of the original attributes had natural bins; only age and income had to be defined for binning.

This transformation was completed in order to utilize the shopping basket algorithm utility provided by the “arules” and “arulesViz” libraries. Using a support of 0.03 and a confidence of 0.8, a total of 730 rules were discovered. The top ten rules, sorted by lift, indicated frequent relationships between age and segment. Often, income or the number of children would be included in the given portion of the rule.

Of the 730 rules discovered in the dataset, none of the rules presented in a given that resulted in defining whether an individual would subscribe or not.

# Appendix - R

rm(list=ls())

library(readr)

#read/import csv file into R

CableTVSubscribersData <- read\_csv("CableTVSubscribersData.csv")

#determine number of rows

nrow(CableTVSubscribersData)

#nrow < 2500; no volume restriction needed

summary(CableTVSubscribersData)

#summary shows no NA values

#double check by running na.omit and recounting rows

CableTVSubscriber2 <- na.omit(CableTVSubscribersData)

nrow(CableTVSubscriber2)

#new data is same as original data

#no NA values confirmed

#use original CableTVSubscriberData for further calculations

#percentages for character types

#check factors

factor(CableTVSubscribersData$gender)

#returns levels Male Female

factor(CableTVSubscribersData$ownHome)

#returns ownNo ownYes

factor(CableTVSubscribersData$subscribe)

#returns subNo subYes

#subNo subYes will be our on/off end goal for categories

factor(CableTVSubscribersData$Segment)

#returns "Moving up" "Suburb mix" "Travelers" "Urban hip"

library(expss)

#run percentages

Percent\_Male <- count\_if("Male", CableTVSubscribersData$gender)/nrow(CableTVSubscribersData)

Percent\_Male

Percent\_Female <- 1 - Percent\_Male

Percent\_Female

Percent\_Own <- count\_if("ownYes", CableTVSubscribersData$ownHome)/nrow(CableTVSubscribersData)

Percent\_Own

Percent\_OwnNo <- 1 - Percent\_Own

Percent\_OwnNo

Percent\_Sub <- count\_if("subYes", CableTVSubscribersData$subscribe)/nrow(CableTVSubscribersData)

Percent\_Sub

Percent\_SubNo <- 1 - Percent\_Sub

Percent\_SubNo

Percent\_SegMovUp <- count\_if("Moving up", CableTVSubscribersData$Segment)/nrow(CableTVSubscribersData)

Percent\_SegSuburb <- count\_if("Suburb mix", CableTVSubscribersData$Segment)/nrow(CableTVSubscribersData)

Percent\_SegTrav <- count\_if("Travelers", CableTVSubscribersData$Segment)/nrow(CableTVSubscribersData)

Percent\_SegUrban <- count\_if("Urban hip", CableTVSubscribersData$Segment)/nrow(CableTVSubscribersData)

Percent\_SegMovUp

Percent\_SegSuburb

Percent\_SegTrav

Percent\_SegUrban

CableTVSubscriber3 <- CableTVSubscriber2

#assign numbers to levels

#1 = moving up

#2 = suburb mix

#3 = travelers

#4 = urban hip

group\_seg <- factor(CableTVSubscriber3$Segment)

#group\_seg -> moving up = 1, suburb mix = 2, travelers = 3, urban hip = 4

levels(group\_seg) <- c(1,2,3,4)

CableTVSubscriber3$Segment <- group\_seg

#begin cluster with age, income, kids, segment

#cannot plot list

library(stats)

library(cluster)

inputs\_age\_inc\_kid\_seg <- c("age", "income", "kids", "Segment")

newCable <- CableTVSubscriber3[inputs\_age\_inc\_kid\_seg]

plot(newCable)

newCable3 <- kmeans(newCable, 3)

newCable4 <- kmeans(newCable, 4)

newCable5 <- kmeans(newCable, 5)

newCable6 <- kmeans(newCable, 6)

newCable7 <- kmeans(newCable, 7)

newCable8 <- kmeans(newCable, 8)

newCable9 <- kmeans(newCable, 9)

newCable10 <- kmeans(newCable, 10)

newCable11 <- kmeans(newCable, 11)

newCable12 <- kmeans(newCable, 12)

newCable13 <- kmeans(newCable, 13)

newCable14 <- kmeans(newCable, 14)

newCable15 <- kmeans(newCable, 15)

newCable16 <- kmeans(newCable, 16)

newCable17 <- kmeans(newCable, 17)

#run elbow method

k\_values <- 3:17

ss\_values <- c(newCable3$tot.withinss, newCable4$tot.withinss, newCable5$tot.withinss, newCable6$tot.withinss,

newCable7$tot.withinss, newCable8$tot.withinss, newCable9$tot.withinss, newCable10$tot.withinss,

newCable11$tot.withinss, newCable12$tot.withinss, newCable13$tot.withinss, newCable14$tot.withinss,

newCable15$tot.withinss, newCable16$tot.withinss, newCable17$tot.withinss)

plot(k\_values, ss\_values, type="b", frame=FALSE, main = "Elbow Method", xlab = "Number of Cluster (k)", ylab = "Total within cluster SS")

#all the pretty colors - plotting clusters

#run one at a time

plot(newCable, col=newCable3$cluster)

plot(newCable, col=newCable4$cluster)

plot(newCable, col=newCable5$cluster)

plot(newCable, col=newCable6$cluster)

plot(newCable, col=newCable7$cluster)

plot(newCable, col=newCable8$cluster)

plot(newCable, col=newCable9$cluster)

plot(newCable, col=newCable10$cluster)

plot(newCable, col=newCable11$cluster)

plot(newCable, col=newCable12$cluster)

plot(newCable, col=newCable13$cluster)

plot(newCable, col=newCable14$cluster)

plot(newCable, col=newCable15$cluster)

plot(newCable, col=newCable16$cluster)

plot(newCable, col=newCable17$cluster)

#silhouette method

sil\_newCable3 <- mean(silhouette(newCable3$cluster, dist(newCable))[,3])

sil\_newCable4 <- mean(silhouette(newCable4$cluster, dist(newCable))[,3])

sil\_newCable5 <- mean(silhouette(newCable5$cluster, dist(newCable))[,3])

sil\_newCable6 <- mean(silhouette(newCable6$cluster, dist(newCable))[,3])

sil\_newCable7 <- mean(silhouette(newCable7$cluster, dist(newCable))[,3])

sil\_newCable8 <- mean(silhouette(newCable8$cluster, dist(newCable))[,3])

sil\_newCable9 <- mean(silhouette(newCable9$cluster, dist(newCable))[,3])

sil\_newCable10 <- mean(silhouette(newCable10$cluster, dist(newCable))[,3])

sil\_newCable11 <- mean(silhouette(newCable11$cluster, dist(newCable))[,3])

sil\_newCable12 <- mean(silhouette(newCable12$cluster, dist(newCable))[,3])

sil\_newCable13 <- mean(silhouette(newCable13$cluster, dist(newCable))[,3])

sil\_newCable14 <- mean(silhouette(newCable14$cluster, dist(newCable))[,3])

sil\_newCable15 <- mean(silhouette(newCable15$cluster, dist(newCable))[,3])

sil\_newCable16 <- mean(silhouette(newCable16$cluster, dist(newCable))[,3])

sil\_newCable17 <- mean(silhouette(newCable17$cluster, dist(newCable))[,3])

silhouette\_values <- c(sil\_newCable3, sil\_newCable4, sil\_newCable5, sil\_newCable6, sil\_newCable7,

sil\_newCable8, sil\_newCable9, sil\_newCable10, sil\_newCable11, sil\_newCable12,

sil\_newCable13, sil\_newCable14, sil\_newCable15, sil\_newCable16, sil\_newCable17)

plot(k\_values, silhouette\_values, type = "b", frame = FALSE, main = "Silhouette Method", xlab = "Number of Cluster (k)", ylab = "Silhouette Values")

#appears that 3 clusters is best using this method

#best fit using between/within

ratioNewCable3 <- newCable3$betweenss/newCable3$tot.withinss

ratioNewCable4 <- newCable4$betweenss/newCable4$tot.withinss

ratioNewCable5 <- newCable5$betweenss/newCable5$tot.withinss

ratioNewCable6 <- newCable6$betweenss/newCable6$tot.withinss

ratioNewCable7 <- newCable7$betweenss/newCable7$tot.withinss

ratioNewCable8 <- newCable8$betweenss/newCable8$tot.withinss

ratioNewCable9 <- newCable9$betweenss/newCable9$tot.withinss

ratioNewCable10 <- newCable10$betweenss/newCable10$tot.withinss

ratioNewCable11 <- newCable11$betweenss/newCable11$tot.withinss

ratioNewCable12 <- newCable12$betweenss/newCable12$tot.withinss

ratioNewCable13 <- newCable13$betweenss/newCable13$tot.withinss

ratioNewCable14 <- newCable14$betweenss/newCable14$tot.withinss

ratioNewCable15 <- newCable15$betweenss/newCable15$tot.withinss

ratioNewCable16 <- newCable16$betweenss/newCable16$tot.withinss

ratioNewCable17 <- newCable17$betweenss/newCable17$tot.withinss

ratioNewCable3

ratioNewCable4

ratioNewCable5

ratioNewCable6

ratioNewCable7

ratioNewCable8

ratioNewCable9

ratioNewCable10

ratioNewCable11

ratioNewCable12

ratioNewCable13

ratioNewCable14

ratioNewCable15

ratioNewCable16

ratioNewCable17

#ratio calls for 3 clusters

#create information for shopping basket rules -> may not be useful

#create binary kids vectors

kids0 <- ifelse(CableTVSubscriber2$kids == 0, 1,0)

kids1 <- ifelse(CableTVSubscriber2$kids == 1, 1,0)

kids2 <- ifelse(CableTVSubscriber2$kids == 2, 1,0)

kids3 <- ifelse(CableTVSubscriber2$kids == 3, 1,0)

kids4 <- ifelse(CableTVSubscriber2$kids == 4, 1,0)

kids5 <- ifelse(CableTVSubscriber2$kids == 5, 1,0)

kids6 <- ifelse(CableTVSubscriber2$kids == 6, 1,0)

kids7 <- ifelse(CableTVSubscriber2$kids == 7, 1,0)

#create kids matrix for binding later

kidsmatrix <- cbind(kids0, kids1, kids2, kids3, kids4, kids5, kids6, kids7)

#create binary from binary

#male = 1, female = 0

#ownYes = 1, no = 0

#subscribeYes = 1, no = 0

group\_Male <- ifelse(CableTVSubscriber2$gender == "Male",1,0)

group\_Female <- ifelse(CableTVSubscriber2$gender == "Feale",1,0)

group\_ownYes <- ifelse(CableTVSubscriber2$ownHome == "ownYes",1,0)

group\_ownNo <- ifelse(CableTVSubscriber2$ownHome == "ownNo",1,0)

group\_subsYes <- ifelse(CableTVSubscriber2$subscribe == "subYes",1,0)

group\_subsNo <- ifelse(CableTVSubscriber2$subscribe == "subNo",1,0)

#create matrix for later

groupmatrix <- cbind(group\_Male, group\_Female, group\_ownYes, group\_ownNo, group\_subsYes, group\_subsNo)

#create binary from segment

group\_seg\_MovingUp <- ifelse(CableTVSubscriber2$Segment == "Moving up",1,0)

group\_seg\_SubMix <- ifelse(CableTVSubscriber2$Segment == "Suburb mix",1,0)

group\_seg\_Travelers <- ifelse(CableTVSubscriber2$Segment == "Travelers",1,0)

group\_seg\_UrbanHip <- ifelse(CableTVSubscriber2$Segment == "Urban hip",1,0)

#create matrix for later

segmatrix <- cbind(group\_seg\_MovingUp, group\_seg\_SubMix, group\_seg\_Travelers, group\_seg\_UrbanHip)

#create bins for binary age

ageUnder25 <- ifelse(CableTVSubscriber2$age <= 25, 1,0)

age25to35 <- ifelse(CableTVSubscriber2$age >25 & CableTVSubscriber2$age <= 35, 1,0)

age35to45 <- ifelse(CableTVSubscriber2$age >35 & CableTVSubscriber2$age <= 45, 1,0)

age45to55 <- ifelse(CableTVSubscriber2$age >45 & CableTVSubscriber2$age <= 55, 1,0)

age55to65 <- ifelse(CableTVSubscriber2$age >55 & CableTVSubscriber2$age <= 65, 1,0)

ageOver65 <- ifelse(CableTVSubscriber2$age > 65, 1,0)

#age matrix

agematrix <- cbind(ageUnder25, age25to35, age35to45, age45to55, age55to65, ageOver65)

#create bins for binary income

incomeUnder10K <- ifelse(CableTVSubscriber2$income <=10000,1,0)

income10to40K <- ifelse(CableTVSubscriber2$income >10000 & CableTVSubscriber2$income <= 40000,1,0)

income40to70K <- ifelse(CableTVSubscriber2$income >40000 & CableTVSubscriber2$income <= 70000,1,0)

income70to100K <- ifelse(CableTVSubscriber2$income >70000 & CableTVSubscriber2$income <= 100000,1,0)

incomeOver100K <- ifelse(CableTVSubscriber2$income >100000,1,0)

#income matrix

incomeMatrix <- cbind(incomeUnder10K, income10to40K, income40to70K, income70to100K, incomeOver100K)

library(arules)

library(arulesViz)

matrixCableSubscriber <- cbind(groupmatrix, agematrix, incomeMatrix, segmatrix, kidsmatrix)

logicalBasket <- apply(matrixCableSubscriber,2,as.logical)

matrixCableSubRules <- as(as.matrix(logicalBasket), "transactions")

#run shopping basket

summary(matrixCableSubRules)

rules <- apriori(matrixCableSubRules, parameter = list(support = .03, confidence = .8))

#manipulate support & confidence as desired

rules

inspect(head(rules, n=10, by = "lift"))

#apply rules to dataframe to sort by RHS -> easy to read results of givens

rules\_info <-

data.frame(

LHS = labels(lhs(rules)),

RHS = labels(rhs(rules)),

quality(rules),

stringsAsFactors = FALSE

)

rules\_info[order(rules\_info$RHS),] #change after $ to column title to reorder

#subscriberYes rules

subscriberYesRules <- rules\_info[rules\_info$RHS == "{group\_subYes}"]

#there are no rules for Yes subscribers (no rules for no Subscribers, either)