Final_Exam

```
library(ggplot2)
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve
3WBa
```

```
overview_of_customer_orders_raw = read.csv("customerdata.csv")

# Renaming the colnames
colnames(overview_of_customer_orders_raw)=c("Cust_Id","Time_for_the_first_order","Fre
quently_Order_DateTime","All_of_the_orders","Last_7_Days_orders","In_the_Last_4_weeks
_orders","Total_Amount","Amount_in_the_Last_7_days","Amount_during_theLast_4_weeks","
Distance_Fromthe_Resturant_on_Average","Typically_DeliveryTime")

#str Data indicates the type of data being kept,this indicates that it is handling da
te as a factor,requiring conversion to Date format

str(overview_of_customer_orders_raw)
```

```
## 'data.frame':    10000 obs. of  11 variables:
##  $ Cust_Id                              : int  1269647 167631 301524 1268254 35716
1 1294857 387095 785080 1288527 1111111 ...
##  $ Time_for_the_first_order             : chr  "6/29/15 10:57" "7/4/15 15:39" "6/2
6/15 9:56" "7/1/15 1:51" ...
##  $ Frequently_Order_DateTime            : chr  "12/10/15 2:18" "12/15/15 14:42" "1
2/9/15 20:45" "12/14/15 1:43" ...
##  $ All_of_the_orders                    : int  212 211 189 184 182 171 168 160 160
158 ...
##  $ Last_7_Days_orders                   : int  6 8 9 6 4 8 13 NA 7 1 ...
##  $ In_the_Last_4_weeks_orders           : int  43 19 33 37 23 27 43 25 40 28 ...
##  $ Total_Amount                         : int  138808 56404 36020 32489 85150 5559
7 19055 39588 4343 15279 ...
##  $ Amount_in_the_Last_7_days            : int  4291 1925 1772 975 1738 1710 1231 0
215 94 ...
##  $ Amount_during_theLast_4_weeks        : int  26853 4177 6404 7110 9958 8436 4014
6705 1060 3336 ...
##  $ Distance_Fromthe_Resturant_on_Average: num  1.6 2.2 2.5 3.1 2.4 1.6 2.1 1.8 2.1
2.1 ...
##  $ Typically_DeliveryTime               : int  51 42 57 55 36 31 48 16 49 54 ...
```

```
dim(overview_of_customer_orders_raw)
```

```
## [1] 10000    11
```

Iam assuming that the average distance from the restaurant and the average delivery time apply to all of the customer orders.

Because of the Date time which was stored in Factor format,we will convert it to Date format in the Following steps

# Now we look at summary data to check if there are any missing values and to have better understanding of the data’s dispersion

```
# summary help us to understand the distribution of each data set as well as any miss
ing values
summary(overview_of_customer_orders_raw)
```

```
##     Cust_Id        Time_for_the_first_order Frequently_Order_DateTime
##  Min.   :     28   Length:10000             Length:10000
##  1st Qu.: 336515   Class :character         Class :character
##  Median : 668340   Mode  :character         Mode  :character
##  Mean   : 671402
##  3rd Qu.:1005002
##  Max.   :1355445
##
##  All_of_the_orders Last_7_Days_orders In_the_Last_4_weeks_orders
##  Min.   :  1.000   Min.   : 1.000     Min.   : 1.000
##  1st Qu.:  1.000   1st Qu.: 1.000     1st Qu.: 1.000
##  Median :  2.000   Median : 1.000     Median : 2.000
##  Mean   :  7.006   Mean   : 1.735     Mean   : 3.198
##  3rd Qu.:  7.000   3rd Qu.: 2.000     3rd Qu.: 4.000
##  Max.   :212.000   Max.   :14.000     Max.   :46.000
##                    NA's   :8077       NA's   :5659
##   Total_Amount    Amount_in_the_Last_7_days Amount_during_theLast_4_weeks
##  Min.   :     1   Min.   :    0.0           Min.   :    0.0
##  1st Qu.:   279   1st Qu.:    0.0           1st Qu.:    0.0
##  Median :   688   Median :    0.0           Median :    0.0
##  Mean   :  2253   Mean   :  109.5           Mean   :  455.5
##  3rd Qu.:  2040   3rd Qu.:    0.0           3rd Qu.:  398.0
##  Max.   :138808   Max.   :10150.0           Max.   :26853.0
##
##  Distance_Fromthe_Resturant_on_Average Typically_DeliveryTime
##  Min.   :-0.800                         Min.   :15.00
##  1st Qu.: 1.700                         1st Qu.:26.00
##  Median : 2.400                         Median :36.50
##  Mean   : 2.356                         Mean   :36.91
##  3rd Qu.: 3.025                         3rd Qu.:47.00
##  Max.   : 5.900                         Max.   :83.00
##
```

Data cleaning and new column creation

I modified the date format,because we do not have all of the details for all orders,I removed the time data for the first and last order

```
overview_of_customer_orders_raw$First_Order_Date= as.Date(overview_of_customer_orders_raw$Time_for_the_first_order,format= "%m/%d/%y")
overview_of_customer_orders_raw$Frequently_Order_Date=as.Date(overview_of_customer_orders_raw$Frequently_Order_DateTime,format = "%m/%d/%y")

overview_of_customer_orders_raw$Present_Date= max(overview_of_customer_orders_raw$Frequently_Order_Date)+ 1

overview_of_customer_orders_raw$countdown_to_the_Last_Order = as.numeric(overview_of_customer_orders_raw$Present_Date - overview_of_customer_orders_raw$Frequently_Order_Date)

overview_of_customer_orders_raw$Days_since_Initial_Order = as.numeric(overview_of_customer_orders_raw$Present_Date - overview_of_customer_orders_raw$First_Order_Date)
```

#Then over the last 7 days and 4 weeks I filtered by cases where the order value was NA and from those users,I removed the users with the shortest time

```
Null_order_7Days = overview_of_customer_orders_raw [ is.na(overview_of_customer_orders_raw$Last_7_Days_orders),]
Null_order_4Weeks = overview_of_customer_orders_raw[is.na(overview_of_customer_orders_raw$In_the_Last_4_weeks_orders),]

print(paste("For Users who had NA value in last 7 Days orders , the minimum value for Recent Order placed is ",min(Null_order_7Days$countdown_to_the_Last_Order),paste("Days"),sep = ""))
```

```
## [1] "For Users who had NA value in last 7 Days orders , the minimum value for Recent Order placed is 9Days"
```

```
print(paste("For users who had NA value in last 4 Week orders,the minimum value for Recent Order placed is",min(Null_order_4Weeks$countdown_to_the_Last_Order),paste("Days"),sep=""))
```

```
## [1] "For users who had NA value in last 4 Week orders,the minimum value for Recent Order placed is30Days"
```

#over the last 7 days and 4 weeks I filtered by cases where the order value was NA and from those users,I removed the users with the shortest time

The minimum days for recent orders are

larger than 7 days and 28 days,respectively.As a result,we may fairly assume that the NA values are not missing,but rather zero.so we gona replace them with 0 .

```
overview_of_customer_orders_raw$Last_7_Days_orders=ifelse(is.na(overview_of_customer_orders_raw$Last_7_Days_orders),0,overview_of_customer_orders_raw$Last_7_Days_orders)

overview_of_customer_orders_raw$In_the_Last_4_weeks_orders=ifelse(is.na(overview_of_customer_orders_raw$In_the_Last_4_weeks_orders),0,overview_of_customer_orders_raw$In_the_Last_4_weeks_orders)
```

I established an average order value(AV) column,which will be used in place of the over all order value and the distance from the restaurant on a average is negative i labled them as 0.

```
overview_of_customer_orders_raw$Distance_Fromthe_Resturant_on_Average =ifelse(overview_of_customer_orders_raw$Distance_Fromthe_Resturant_on_Average<0,0,overview_of_customer_orders_raw$Distance_Fromthe_Resturant_on_Average)


overview_of_customer_orders_raw$Av_All =round(overview_of_customer_orders_raw$Total_Amount/overview_of_customer_orders_raw$All_of_the_orders,0)

overview_of_customer_orders_raw$Av_Last_7_Days =round(ifelse(overview_of_customer_orders_raw$Last_7_Days_orders==0,0,overview_of_customer_orders_raw$Amount_in_the_Last_7_days/overview_of_customer_orders_raw$Last_7_Days_orders),0)

overview_of_customer_orders_raw$Av_Last_4_Weeks =round(ifelse(overview_of_customer_orders_raw$In_the_Last_4_weeks_orders==0,0,overview_of_customer_orders_raw$Amount_during_theLast_4_weeks/overview_of_customer_orders_raw$In_the_Last_4_weeks_orders),0)
```

Customer segmentation

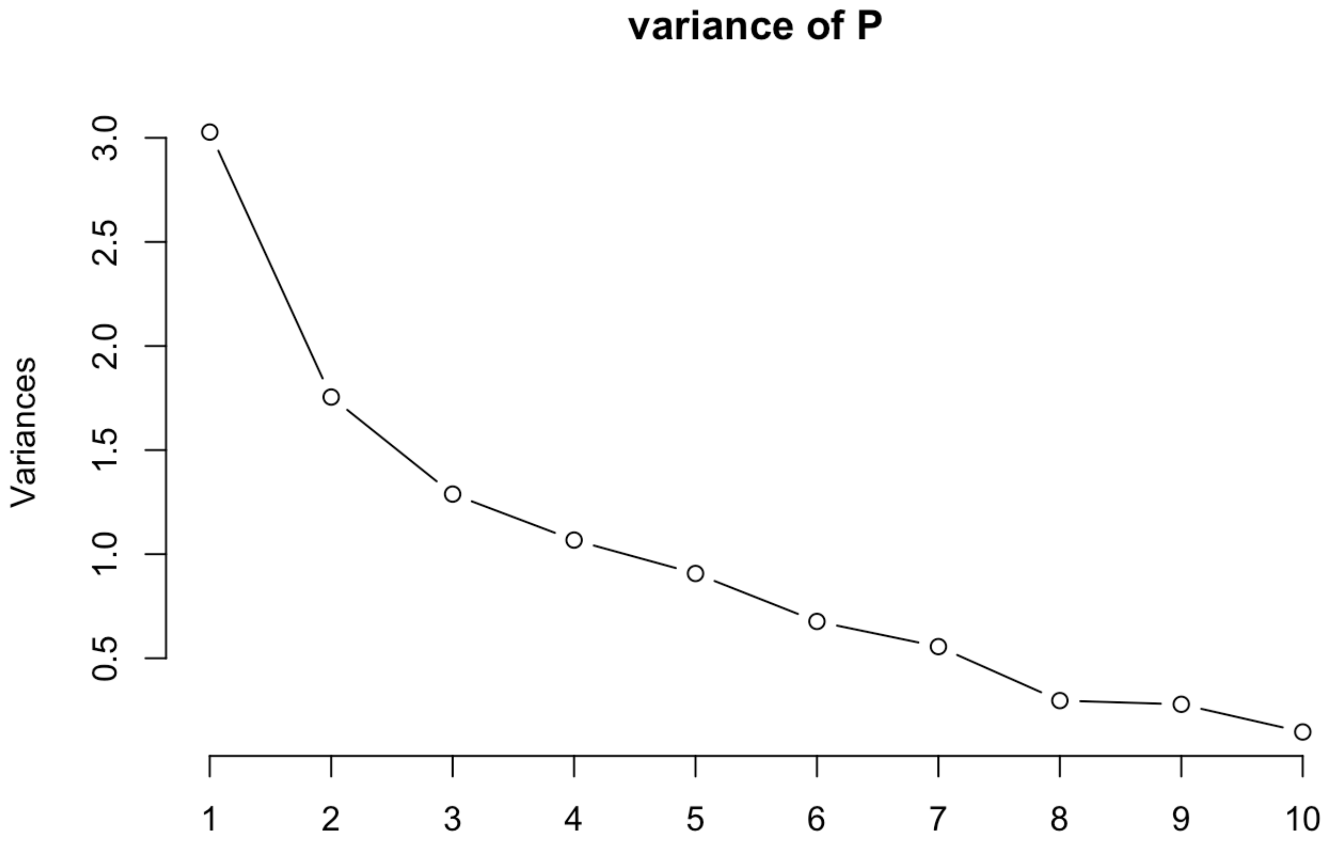
```
q1 = 100 - round(100*sum(overview_of_customer_orders_raw$Last_7_Days_orders==0)/nrow(
overview_of_customer_orders_raw),0)

q2 = 100 - round(100*sum(overview_of_customer_orders_raw$In_the_Last_4_weeks_orders==
0)/nrow(overview_of_customer_orders_raw),0)
```

In the “q1” percent of consumers transacted in the previous 7 days ,while “q2” percent transacted in the previous 4 weeks.This indicates that we have a large number of users that have not interacted in the last month.

I am generating a filtered data set for our raw data that only takes in to account relevant columns while building the model.Ordercount,Av,Distance from the restaurant on Average,Typically deliverytime since first and last orders are the key columns for our analysis.we have essentially deleted a few columns that provide redundant data,such as Total value of the order which is a function of total orders and AV

Applying pricipal component analysis

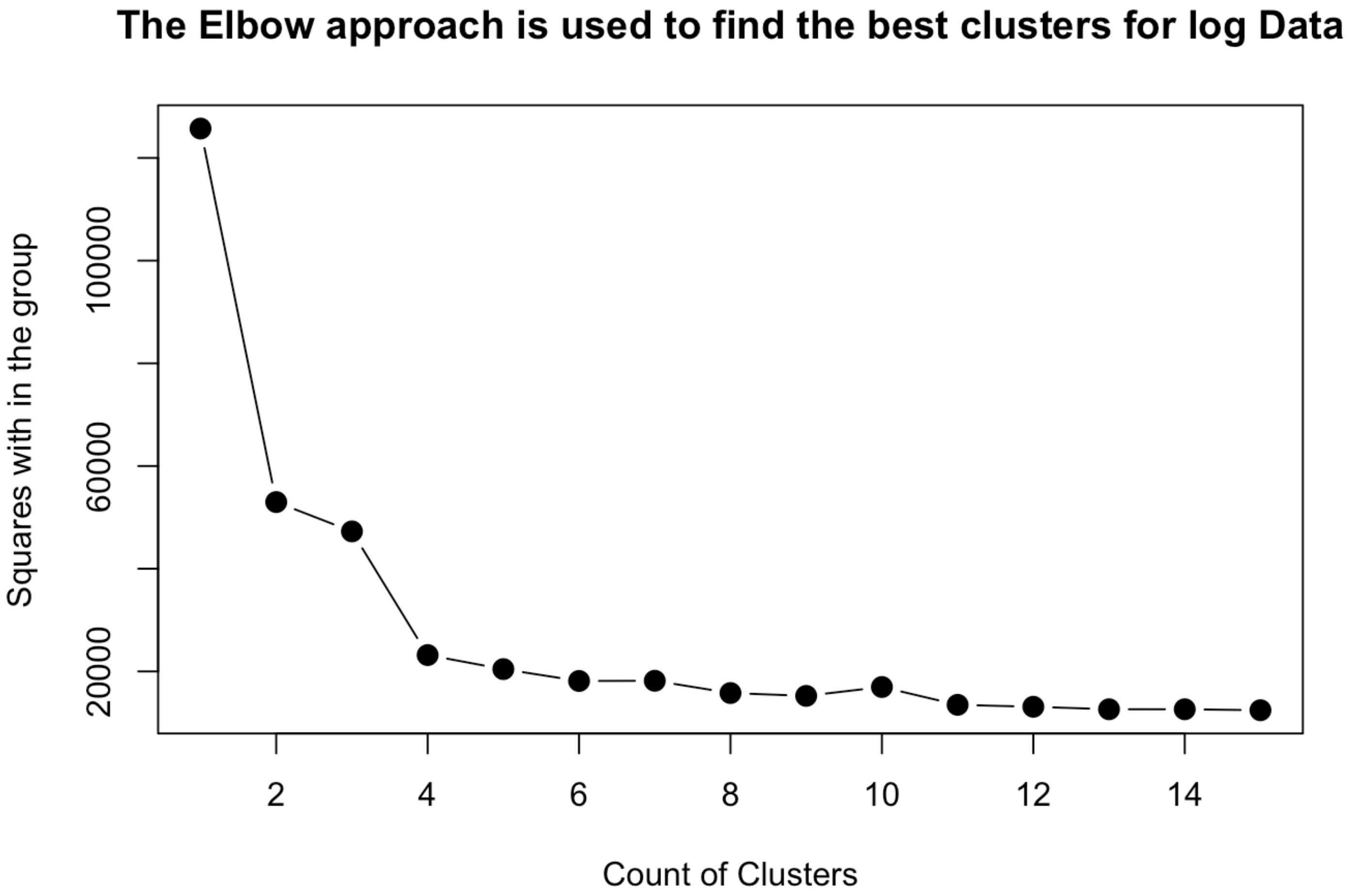
```
f_data = overview_of_customer_orders_raw[ , c(1,4:6,10,11,15:19)]
set.seed(1234)

p_data1 = prcomp(f_data[,-1],center = T,scale. = T)

plot(p_data1,type = "l",
     main = "variance of P")
```



when we look at the fifth and the sixth principal components,we can see that the variance still substantial.so lets get started with all of the variables in this model

Here I’m attempting to construct a clustering model in order to determine whether we can separate people into distinct catergories.Using k means clustering for this.i opted to look at the log and norm transformations and use the elbow technique to compute the minimum errors and decide how many clusters to divide the data into because the summary data comprises variables at different scales.

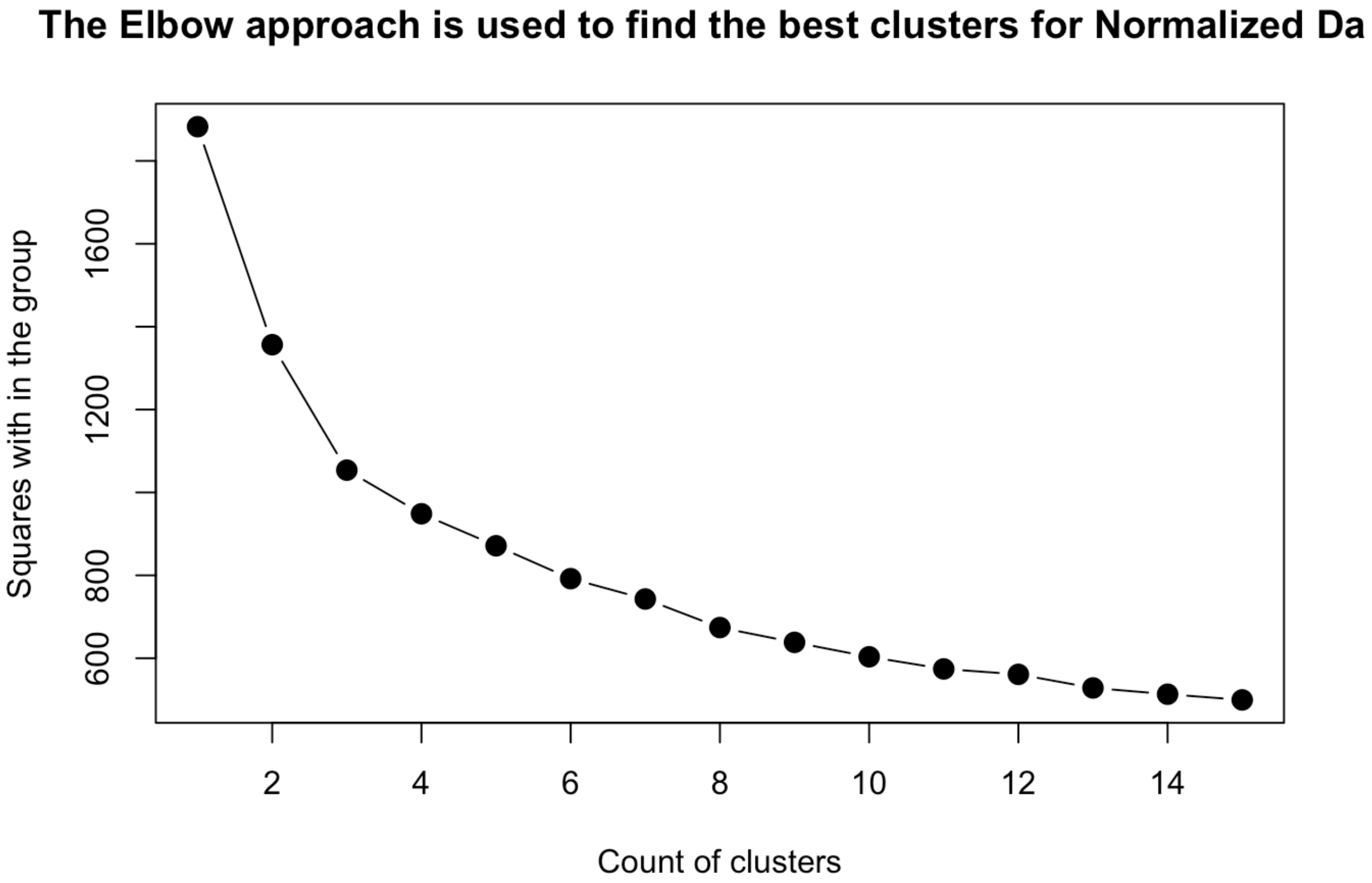
```
set.seed(00909)

normalize <-function(x) {
  return((x-min(x))/(max(x) - min(x)))

}
ft_data_log = log(f_data[,-1]+2)

ft_data_norm = as.data.frame(lapply(f_data[,-1],normalize))


score_wss_log<-(nrow(ft_data_log)-1)*sum(apply(ft_data_log,2,var))

for (i in 2:15)  score_wss_log[i] <-sum(kmeans(ft_data_log,centers = i)$withinss)



plot(1:15,score_wss_log[1:15],type = "b",xlab = "Count of Clusters",ylab="Squares wit
h in the group",main = "The Elbow approach is used to find the best clusters for log 
Data",pch=20,cex=2)
```



```
score_wss_normal<-(nrow(ft_data_norm)-1)*sum(apply(ft_data_norm,2,var))

for (i in 2:15)
  score_wss_normal[i] <-sum(kmeans(ft_data_norm,
                                   centers = i)$withinss)
```

```
## Warning: did not converge in 10 iterations
```

```
plot(1:15,score_wss_normal[1:15],type = "b",xlab = "Count of clusters",ylab = "Square
s with in the group",main = "The Elbow approach is used to find the best clusters for
Normalized Data",pch=20,cex=2)
```



The normalized data makes more sense to proceed with, as seen by the charts above.

Denormalize the data and look at the centers means of each variable in the Three clusters

```
minvec <-sapply(f_data[,-1],min)
maxvec<-sapply(f_data[,-1],max)
denormalize<-function(x,minval,maxval)
  return(x*(maxval-minval))



set.seed(009)
kmeans_3_cl_normal = kmeans(ft_data_norm,3,nstart = 100)

kmeans_3_cl_actual = NULL
t1=NULL

for (i in 1:10)
  {  t1 = (kmeans_3_cl_normal$centers[,i] * (maxvec[i]-minvec[i])) + minvec[i]
     kmeans_3_cl_actual = cbind(kmeans_3_cl_actual,t1)

}

colnames(kmeans_3_cl_actual) = colnames(f_data[-1])
print("Below is the mean value of all variables in each cluster")
```

```
## [1] "Below is the mean value of all variables in each cluster"
```

```
kmeans_3_cl_actual
```

```
##   All_of_the_orders Last_7_Days_orders In_the_Last_4_weeks_orders
## 1          2.797698          0.3056266                   1.204859
## 2          2.513654          0.0000000                   0.000000
## 3         17.050763          0.7103517                   3.042468
##   Distance_Fromthe_Resturant_on_Average Typically_DeliveryTime
## 1                              2.355601               36.06829
## 2                              2.378023               38.51625
## 3                              2.338056               36.37691
##   countdown_to_the_Last_Order Days_since_Initial_Order   Av_All Av_Last_7_Days
## 1                    56.89335                 74.20307 377.2760       73.37749
## 2                   142.50163                159.91125 360.9083        0.00000
## 3                    47.51924                173.80425 336.6768      124.55076
##   Av_Last_4_Weeks
## 1        200.5054
## 2          0.0000
## 3        255.7472
```

```
print("The following table shows the number of customers in each cluster")
```

```
## [1] "The following table shows the number of customers in each cluster"
```

```
kmeans_3_cl_normal$size
```

```
## [1] 3910 3076 3014
```

```
#kmeans_3_cl_normal$centers
#fviz_cluster(kmeans_3_cl_normal,data = ft_data_norm)
```

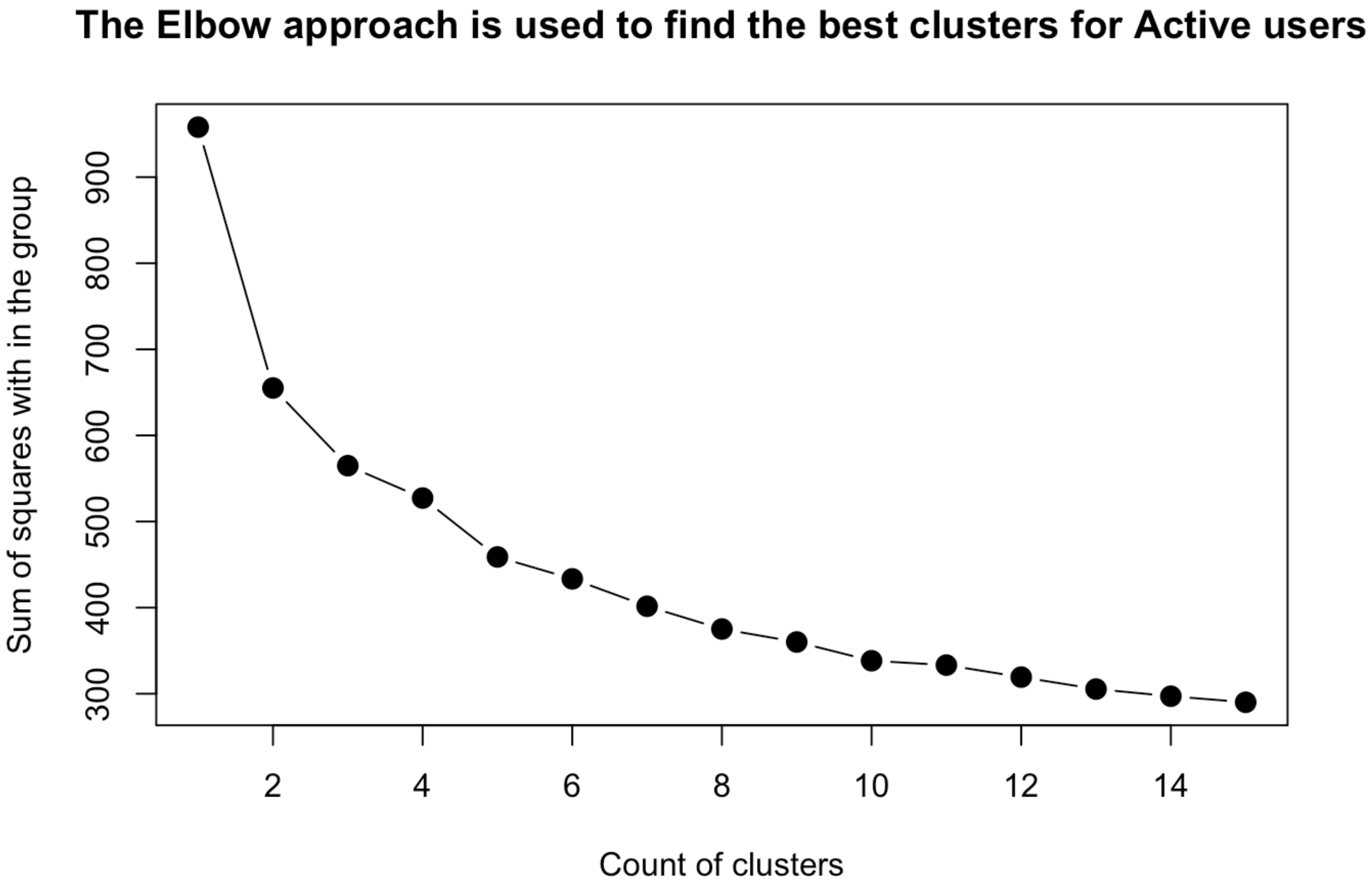
we can observe from the mean values of each variable in each of the three clusters that transaction frequency and AV are the most variable, whereas typical delivery time and distance on average are not.

Classification of customers:

1.Over the last 7 days and 4 weeks customers are active but with Low frequency and Low Av. 2.Over the last 7 days and 4 weeks ,customers are active with high frequency and High Av. 3.Customers are not in active since last 7 days and over 4 weeks

Building the cluster for active customers by removing the users who did not interacted in last 4 weeks

```
ft_order_data = f_data[f_data$In_the_Last_4_weeks_orders !=0,]

normalize <-function(x) {
  return((x-min(x))/ (max(x)-min(x)))
}

 ft_order_data_norm=as.data.frame(lapply(ft_order_data[,-1],normalize))

 score_wss_order_normal<-(nrow(ft_order_data_norm)-1)*sum(apply(ft_order_data_norm,2,
var))

 for (i in 2:15) score_wss_order_normal[i] <- sum(kmeans(ft_order_data_norm,centers =
i)$withinss)




 plot(1:15,score_wss_order_normal[1:15],type = "b", xlab = "Count of clusters", ylab
= "Sum of squares with in the group", main = "The Elbow approach is used to find the
best clusters for Active users",pch=20,cex=2)
```



Given 3 and 4 appear to be the best options,let’s go with 4 to see if we can improve user segment differentation

```
library(ggplot2)
library(cluster)
library(factoextra)

minvec1 <-sapply(ft_order_data[,-1],min)
maxvec1 <-sapply(ft_order_data[,-1],max)
denormalize <-function(x,minval,maxval)
  return(x*(maxval-minval)+ minval)


set.seed(0091)
kmeans_2_cl_nom_order_data=kmeans(ft_order_data_norm,2,nstart = 100)


kmeans_4_cl_nom_order_data= kmeans(ft_order_data_norm,4,nstart = 100)

kmeans_2_cl_act_order_data = NULL
kmeans_4_cl_act_order_data= NULL

test1=NULL

for (i in 1:10)
  {         test1=(kmeans_2_cl_nom_order_data$centers[,i]*(maxvec1[i]-minvec[i]))+ minvec1[i]
  kmeans_2_cl_act_order_data = cbind(kmeans_2_cl_act_order_data,test1)

}

colnames(kmeans_2_cl_act_order_data)=colnames(f_data[-1])


test1=NULL

for (i in 1:10)
  {             test1=(kmeans_4_cl_nom_order_data$centers[,i]*(maxvec1[i]-minvec[i]))+ minvec1[i]
  kmeans_4_cl_act_order_data=cbind(kmeans_4_cl_act_order_data,test1)

}
colnames(kmeans_4_cl_act_order_data) = colnames(f_data[-1])

kmeans_2_cl_act_order_data
```

```
##   All_of_the_orders Last_7_Days_orders In_the_Last_4_weeks_orders
## 1         19.563269          0.9521090                   4.152451
## 2          3.474092          0.5661017                   2.247953
##   Distance_Fromthe_Resturant_on_Average Typically_DeliveryTime
## 1                              2.338752               36.80888
## 2                              2.385908               35.70654
##   countdown_to_the_Last_Order Days_since_Initial_Order   Av_All Av_Last_7_Days
## 1                    38.74868                173.07865 338.9868       167.4262
## 2                    38.88475                 61.14092 366.1361       136.1937
##   Av_Last_4_Weeks
## 1        345.9350
## 2        371.6465
```

```
kmeans_2_cl_nom_order_data$size
```

```
## [1] 2276 2065
```

```
print("Below is the mean value of all variables in each cluster")
```

```
## [1] "Below is the mean value of all variables in each cluster"
```

```
kmeans_4_cl_act_order_data
```

```
##   All_of_the_orders Last_7_Days_orders In_the_Last_4_weeks_orders
## 1          3.257091          0.5141812                   2.139128
## 2          3.699584          0.6122661                   2.314438
## 3         19.138718          1.0131694                   4.219237
## 4         19.862249          0.8988666                   4.117466
##   Distance_Fromthe_Resturant_on_Average Typically_DeliveryTime
## 1                              2.198994               25.93779
## 2                              2.587838               47.22349
## 3                              2.302107               25.57243
## 4                              2.384307               47.60680
##   countdown_to_the_Last_Order Days_since_Initial_Order   Av_All Av_Last_7_Days
## 1                    40.36597                 61.06587 361.3715       119.7832
## 2                    37.48649                 60.74428 367.8285       154.9158
## 3                    37.75417                170.86392 344.2511       172.5979
## 4                    39.49869                174.70619 337.1168       161.9538
##   Av_Last_4_Weeks
## 1        365.2315
## 2        377.2214
## 3        349.2529
## 4        344.3017
```

```
print("The following table shows the number of customers in each cluster")
```

```
## [1] "The following table shows the number of customers in each cluster"
```

```
kmeans_4_cl_nom_order_data$size
```

```
## [1] 1093  962 1139 1147
```

Let’s take a closer look at the outcomes.

1.cluster(1) has a low frequency , a high Average order value ,a long delivery time,and a higher distance from the restaurant. 2.cluster(2) features a high frequency of users,a poor Average oder value and long delivery time and distance from the restaurant is lower 3.cluster(3) has a low frequency ,a low Average order value particularly in the recent 7 days,and a low delivery time, and a lower distance from the restaurant 4.cluster (4) has high frequency , a high Average order value and a low delivery time

so in addition to order frequency and average order value we are seeing delivery

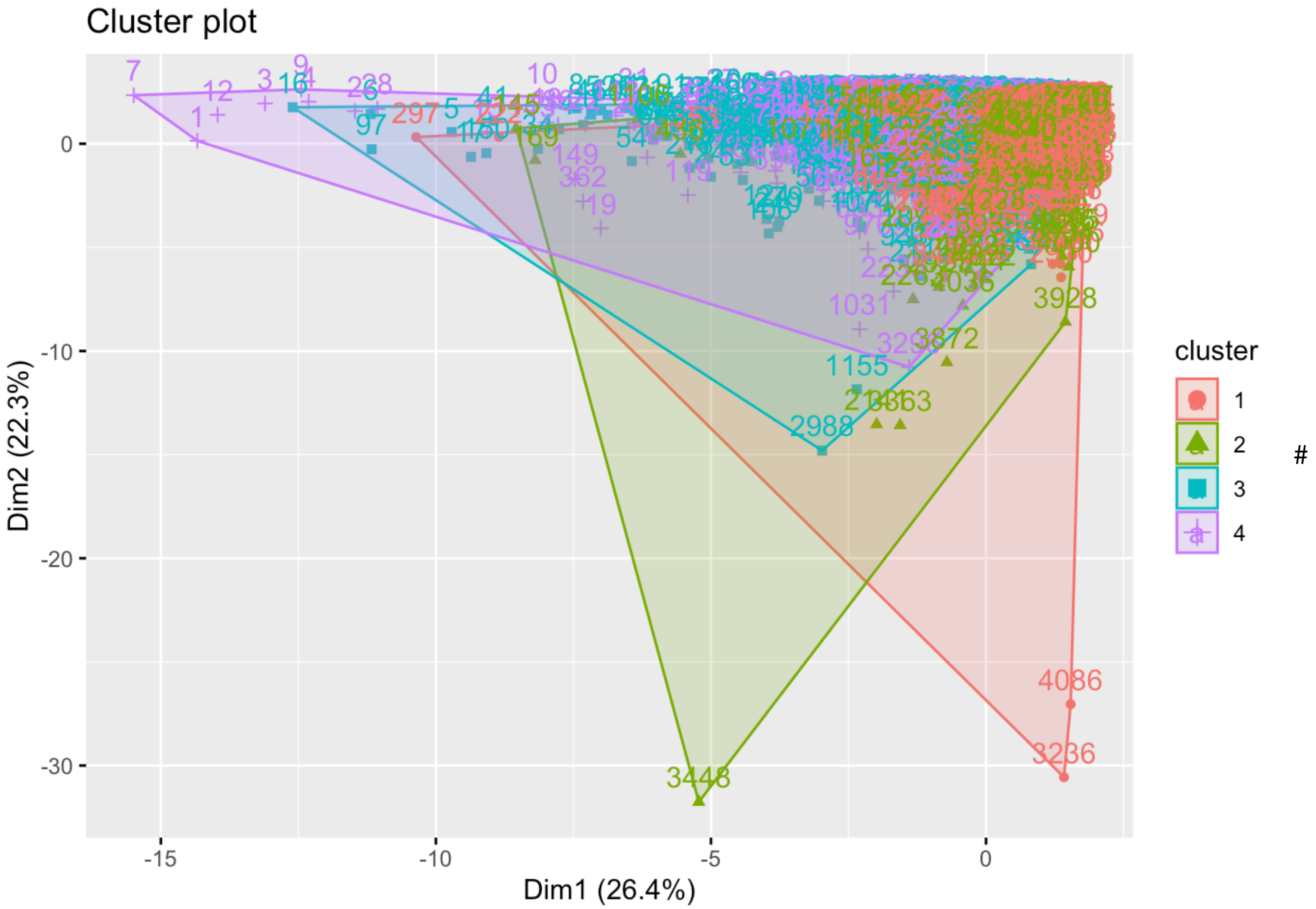
time are to a lesser extent, distance from the restaurant emerge as important variables in creating clusters. And that delivery time and restaurant distance are not directly proportinal,with more data we can find so this is the case .

#At last consider the 4 cluster plot.

```
#kmeans_3_cl_normal$centers
#fviz_cluster(kmeans_2_cl_nom_order_data,data = ft_data_norm)

print("The cluster visualization of four clusters is shown in the graph below")
```

```
## [1] "The cluster visualization of four clusters is shown in the graph below"
```

```
fviz_cluster(kmeans_4_cl_nom_order_data,data = ft_order_data_norm)
```



From the above graph it is difficult to understand the information,but we can see how each cluster has its own different boundaries on the x and y axis, which are distinguishing qualities that divide the users into different groups.

Conclusion

I looked at Food Delivery customer data in the previous analysis .I noticed that the order frequency and values as crucial indicators to cluster users when i looked at all users together,but because of a large portion of them were not active,I only had a look at those who transacted in the previous four weeks to better understand them.A part from the above sentences i also discovered that Average delivery time was another important element for the further users.This investigation can be used to work on early consumer identification and understanding the importance to them while placing the orders which help to better target the customers.For better explanation i can consider the cluster 1 and cluster 2 we find that the average delivery time is not important to the customers and i can also display the restaurants which are at long in distance,and from the cluster 3 and cluster 4 i should provide more choices for quicker delivery.