Assignment

```r
library(plyr)
library(lubridate)
library(sqldf)
library(reshape2)
library(ggplot2)

setwd("C:\\Users\\viddi\\Desktop\\New folder\\R CODE VIDHYADHAR")
data <- read.csv("Web_Baskets_2020.csv",sep=";")
```

- **What kind of products purchased most?**

```r
a <- dcast(data, article_cat + article_name ~ "Freq", value.var="quantity",sum)
head(a[order(-a$Freq),])
```

##	article_cat	article_name	Freq
## 4221	shirts	women shirt	123475
## 557	bedsheet	fitted sheet beaver	60392
## 3203	nightgown/pajamas	women nightgown	46144
## 5111	underpants	women panties	42409
## 4197	shirts	women blouse shirt	36854
## 795	bras	women underwired bra	34565

Answer: Product "women shirt" with shirts category has been purchased most (max frequency)

- **What are the most successful category in our online business?**

```r
a <- dcast(data, article_cat ~ "Freq", value.var="quantity",sum)
head(a[order(-a$Freq),])
```

##	article_cat	Freq
## 164	pants	419885
## 185	shirts	363677
## 229	underpants	278266
## 39	bras	189509
## 147	nightgown/pajamas	176958
## 215	terry goods	132118

Ans: pants category is the most successful (max frequency)

```r
data$d1 <- weekdays(ymd(data$date))

## Warning: 88854 failed to parse.

data$m <- month(ymd(data$date),label=TRUE)
```

```
## Warning: 88854 failed to parse.
```

Please note:

```
unique(data[!complete.cases(data),"date"])

## [1] "2020-04-31"
```

There is an invalid date: 2020-04-31; there can't be 31 days in the month of April

- **Which days of the week do we make the most/least transactions?**

```
b <- dcast(data, d1 ~ "freq", value.var="quantity",sum)
b[order(-b$freq),]

##     d1          freq
## 5   Thursday    635451
## 6   Tuesday     612735
## 7   Wednesday   581510
## 1   Friday      485628
## 3   Saturday    440962
## 4   Sunday      439852
## 2   Monday      344129
## 8   <NA>        99813
```

Ans: Thursday most and Monday least transactions.

- **How are the different articles and categories performing?**

```
b <- dcast(data, article_cat ~ "freq", value.var="quantity",sum)
head(b[order(-b$freq),])

##        article_cat          freq
## 164    pants                419885
## 185    shirts               363677
## 229    underpants           278266
## 39     bras                 189509
## 147    nightgown/pajamas    176958
## 215    terry goods          132118

tail(b[order(-b$freq),])

##      article_cat                                 freq
## 57   coffee expertise articles                   1
## 90   food bowls                                  1
## 122  jewelry cases                               1
## 162  painting and wallpapering accessories       1
## 169  photo & accessories                         1
## 213  tent & accessories                          1
```

Ans: pants are mostly sold and tent & accessories are least sold

```
b <- dcast(data,  article_name ~ "freq", value.var="quantity",sum)
head(b[order(-b$freq),])
```

```
##              article_name            freq
## 4796         women shirt             167017
## 1415         fitted sheet beaver      60392
## 4711         women nightgown          46144
## 4728         women panties            42412
## 4431         women blouse shirt       40792
## 4724         women pajamas            35156
```

```
tail(b[order(-b$freq),])
```

```
##              article_name            freq
## 5070         xl eds pot              1
## 5071         xl knitting needles     1
## 5107         zwilling cheese knife   1
## 5110         zwilling fillet knife   1
## 5113         zwilling knife set      1
## 5117         zwilling pressure cooker 1
```

Ans. Women’s shirt is the most sold and 'Zwilling pressure cooker' is the least sold.

- **1a. Find best-selling category**

```
f <- dcast(data, article_cat + m ~ "freq", value.var="quantity",sum)
p <- sqldf("select article_cat, max(freq) as Freq
from f
group by article_cat
order by Freq desc")
head(p,5)
```

```
##              article_cat             Freq
## 1            pants                   395532
## 2            shirts                  345900
## 3            underpants              264587
## 4            bras                    179899
## 5            nightgown/pajamas       168117
```

Ans.5 best-selling categories: pants, shirts, underpants, bras & nightgown/pajamas

1b. Name the categories and visualize the respective best-selling articles of each of the 5 categories per day in one picture

```
a <- sqldf("select article_cat, max(freq) as Freq
from f
group by article_cat
order by Freq desc")

b <- a[1:5,1]
```

```
df <- data[data$article_cat %in% b,]
df[!duplicated(df$article_cat),"article_cat"]
```

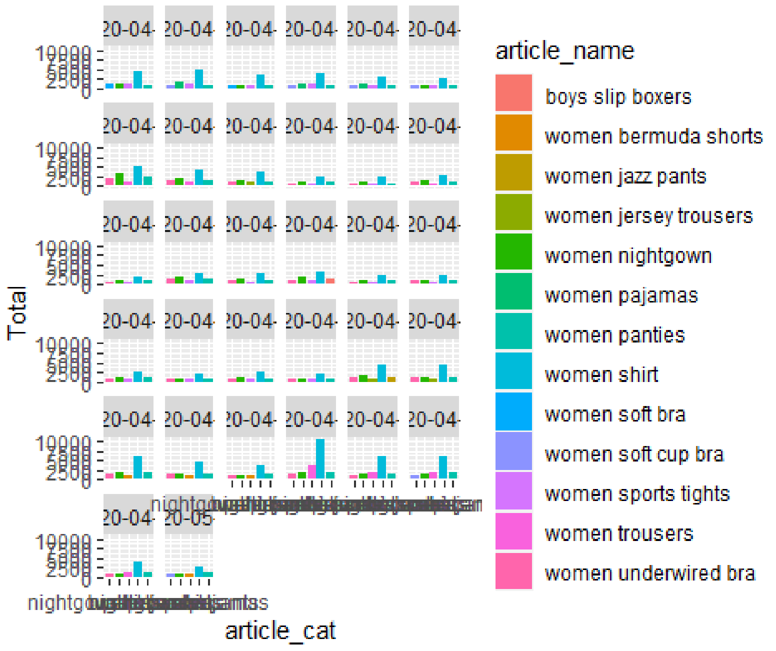
```
## [1] "pants"             "underpants"        "bras"
## [4] "nightgown/pajamas" "shirts"
```

```
p <- dcast(df, date + article_cat+article_name ~ "Freq", value.var="article_name",length)
#head(p)
kane <- sqldf("select date, article_cat, article_name, max(Freq) as Total
from p
group by date, article_cat
")
#head(kane,10)
#head(kane)
kane[!duplicated(kane$date),"date"]
```

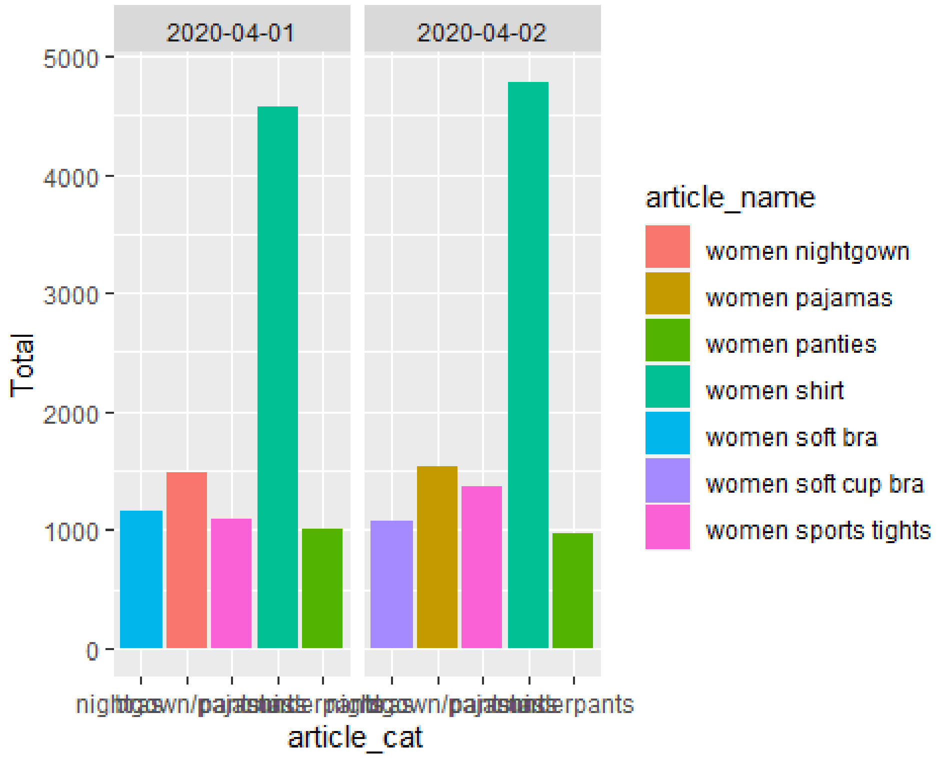
```
##  [1] "2020-04-01" "2020-04-02" "2020-04-03" "2020-04-04" "2020-04-05"
##  [6] "2020-04-06" "2020-04-07" "2020-04-08" "2020-04-09" "2020-04-10"
## [11] "2020-04-11" "2020-04-12" "2020-04-13" "2020-04-14" "2020-04-15"
## [16] "2020-04-16" "2020-04-17" "2020-04-18" "2020-04-19" "2020-04-20"
## [21] "2020-04-21" "2020-04-22" "2020-04-23" "2020-04-24" "2020-04-25"
## [26] "2020-04-26" "2020-04-27" "2020-04-28" "2020-04-29" "2020-04-30"
## [31] "2020-04-31" "2020-05-01"
```

Please note: To fit all the graphs into a single chart is impossible. Here, I have shown only 2 dates, which is working just fine.

```
ggplot(kane, aes(fill=article_name, y=Total, x=article_cat)) +
  geom_bar(position="dodge", stat="identity")+facet_wrap(~date)
```



```r
k <- kane[kane$date %in% c("2020-04-01","2020-04-02"),]
ggplot(k, aes(fill=article_name, y=Total, x=article_cat)) +
    geom_bar(position="dodge", stat="identity")+facet_wrap(~date)
```



Q2 a and b: Extract at least 4 new attributes and for each created variable, provide a description of what it measures

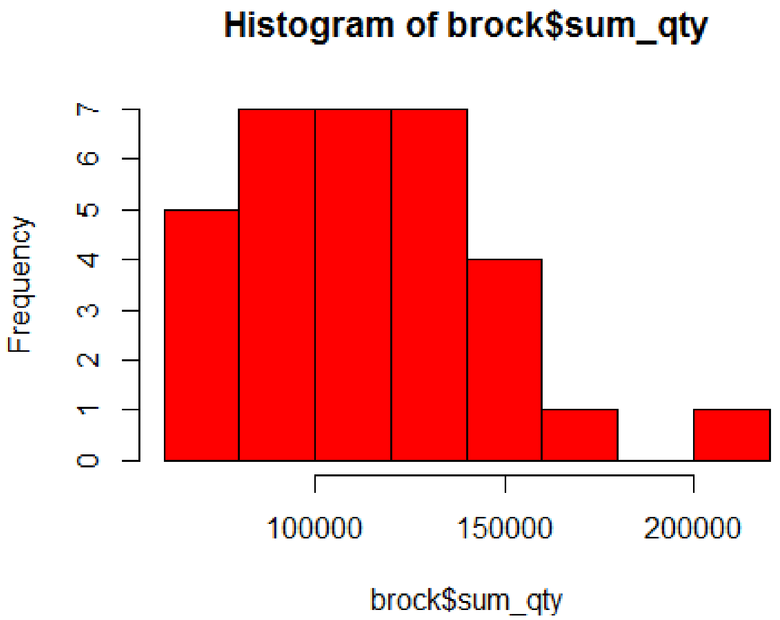
```r
brock <- sqldf("select date, sum(quantity) as sum_qty,
                    avg(quantity) as mean_qty,
                       median(quantity) as mdn_qty,
                   count(quantity) as count_qty
from data
group by date
")
```

	date	sum_qty	mean_qty	mdn_qty	count_qty
1	2020-04-01	127962	1.117084	1	114550
2	2020-04-02	143100	1.108314	1	129115
3	2020-04-03	100536	1.116161	1	90073
4	2020-04-04	113038	1.113346	1	101530
5	2020-04-05	102896	1.125186	1	91448
6	2020-04-06	80854	1.121508	1	72094

Q2.c. Visualize at least 2 different newly created variable distributions and explain the picture

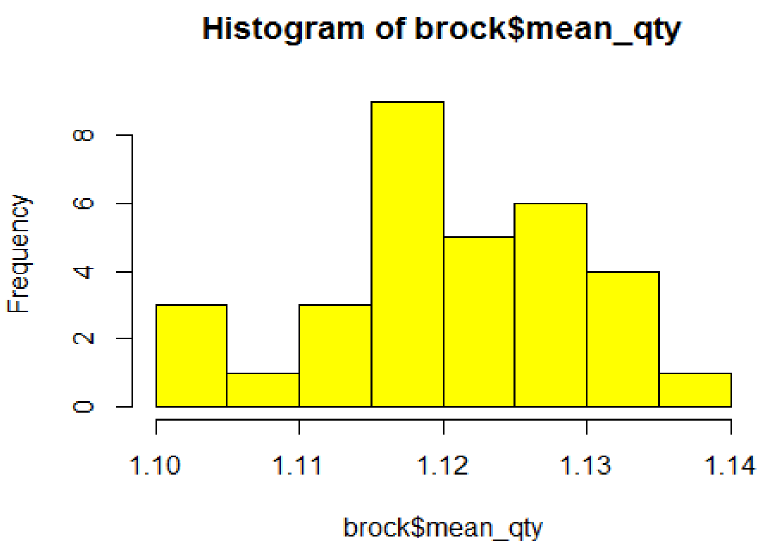
```r
hist(brock$sum_qty, col="red")
```

#It's positively skewd distribution, where we can see few extreme high values



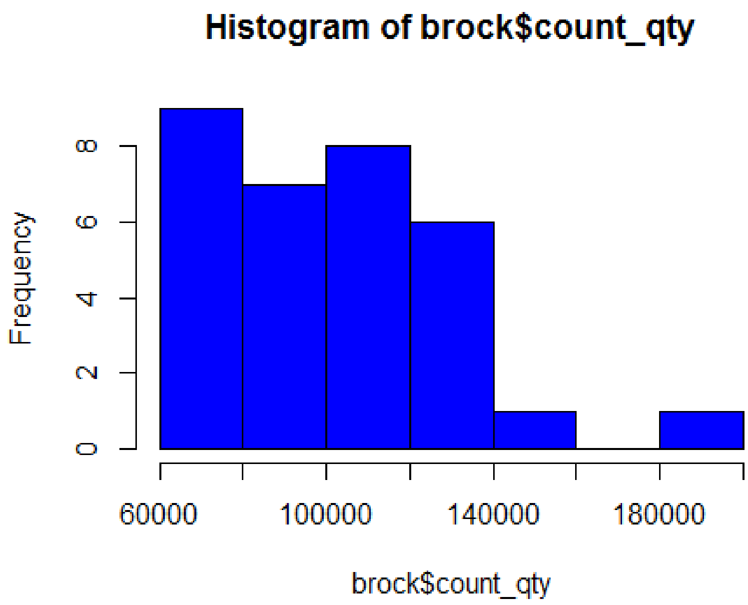

```r
hist(brock$mean_qty, col="yellow")
```

#It's almost a normal distribution, but slightly negatively skewed.



```r
hist(brock$count_qty, col="blue")
```

#It's positively skewd distribution, where we can see few extreme high values



Q2.d. Normalize the variable as input to clustering.

```r
#First remove the redundant variables from the data
brock$date <- NULL
brock$mdn_qty <- NULL

Here we are normalizing the data so that variables with high value

doesn't have a high impact on the clustering

br <- scale(brock)
```

The scale function will automatically normalize all the variables

```
 sum_qty            mean_qty       count_qty
 [1,]  0.44743481 -0.361047826  0.457408351
 [2,]  0.92410662 -1.374675672  0.970305758
 [3,] -0.41616680 -0.467717100 -0.404533964
 [4,] -0.02249848 -0.793134748 -0.001082842
 [5,] -0.34185411  0.575338829 -0.356114195
 [6,] -1.03592203  0.150252192 -1.037653255
 [7,]  1.17308562 -2.021751391  1.245893475
 [8,]  0.07014047 -1.770788339  0.120054616
 [9,] -0.31263285 -2.139408634 -0.257795653
[10,] -1.08750007  0.013615751 -1.086460382
[11,] -1.15205135 -0.115641611 -1.148472900
[12,] -0.68129936 -0.749580136 -0.663817424
[13,] -1.44328782 -0.476768845 -1.433498071
```

```
[14,]  0.57197154 -0.007518471  0.570728218
[15,] -0.15537965  0.735732355 -0.174971438
[16,]  0.15112871 -0.844305882  0.174812973
[17,] -0.99643554 -0.195601257 -0.991381563
[18,] -1.19553687 -0.168936030 -1.190941439
[19,] -0.80051455  0.810312203 -0.816964752
[20,] -1.27337626  0.313091323 -1.277322307
[21,]  0.42640054  1.213606271  0.388000814
[22,] -0.61328436  1.724741818 -0.651738893
[23,]  1.03340296  1.206153311  0.988617234
[24,]  0.72686312  0.906514529  0.695211042
[25,]  1.92773727  1.001450172  1.881759488
[26,]  1.34636644  0.827341735  1.312343006
[27,]  0.26111778  1.374566670  0.219852162
[28,]  2.79505452 -0.514899537  2.815257418
[29,]  0.65248745 -0.098318254  0.654080649
[30,]  0.30391056 -0.557409832  0.319649707
[31,] -0.43893293  0.361662280 -0.447460290
[32,] -0.84453537  1.443124125 -0.873765542
attr(,"scaled:center")
     sum_qty     mean_qty    count_qty
1.137525e+05 1.120208e+00 1.015608e+05
attr(,"scaled:scale")
     sum_qty     mean_qty    count_qty
3.175770e+04 8.652041e-03 2.839749e+04
```

Q3 A.B.C.: Use the extracted information from Q2 and find clusters of articles

```r
library(factoextra)
library(cluster)
library(fpc)
library(NbClust)
library(clValid)
library(magrittr)
library(clustertend)
```

Hopkins's test shows that the data is not suitable for clustering

```r
res <- get_clust_tendency(br, n = nrow(br)-1, graph = FALSE)
res$hopkins_stat

## [1] 0.80341
```

We will apply clustering, nonetheless.

Method I: using silhouette method

```r
nb <- NbClust(br,  distance = "euclidean", min.nc=2, max.nc=15,
method = "kmeans",index = "silhouette")

nb$All.index## maximum value of silhouette shows best number of clusters
```

```
##       2      3      4      5      6      7      8      9     10     11
12
## 0.3763 0.4151 0.4011 0.3757 0.4282 0.4014 0.4090 0.4861 0.4803 0.5219 0
.5163
##     13     14     15
## 0.5078 0.5365 0.5509
```

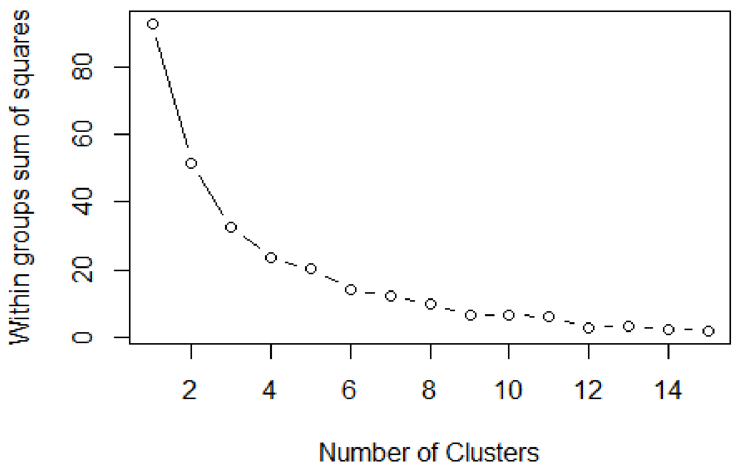
```
nb$Best.nc
```

```
## Number_clusters     Value_Index
##         15.0000          0.5509
```

Sillhoutte suggests to go for 15 clusters, which doesn't seem to be right

Method III: Scree plot to determine the number of clusters

```
wss <- (nrow(br)-1)*sum(apply(br,2,var))
for (i in 2:15) {
  wss[i] <- sum(kmeans(br,centers=i)$withinss)
}
plot(1:15, wss, type="b", xlab="Number of Clusters",ylab="Within groups su
m of squares")
```



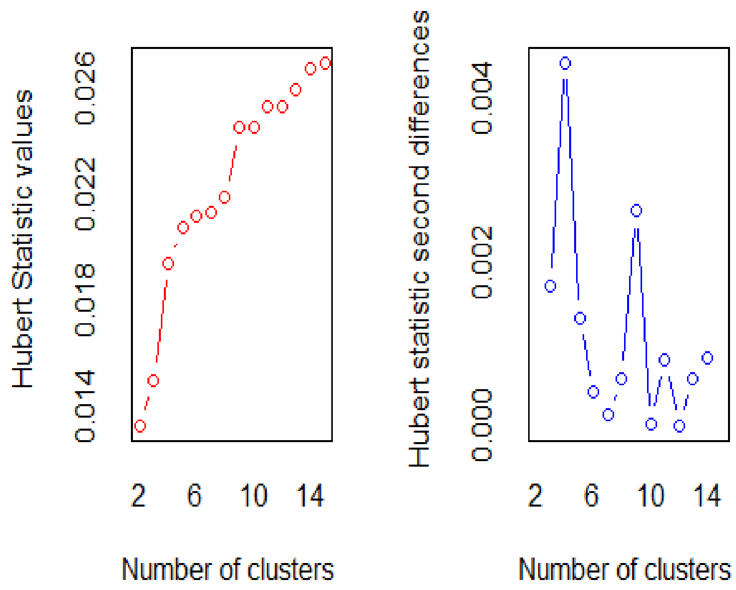
#Scree plot suggests to go for 3 or 4 clusters

Best Method IV: Using all 30 ways of measure

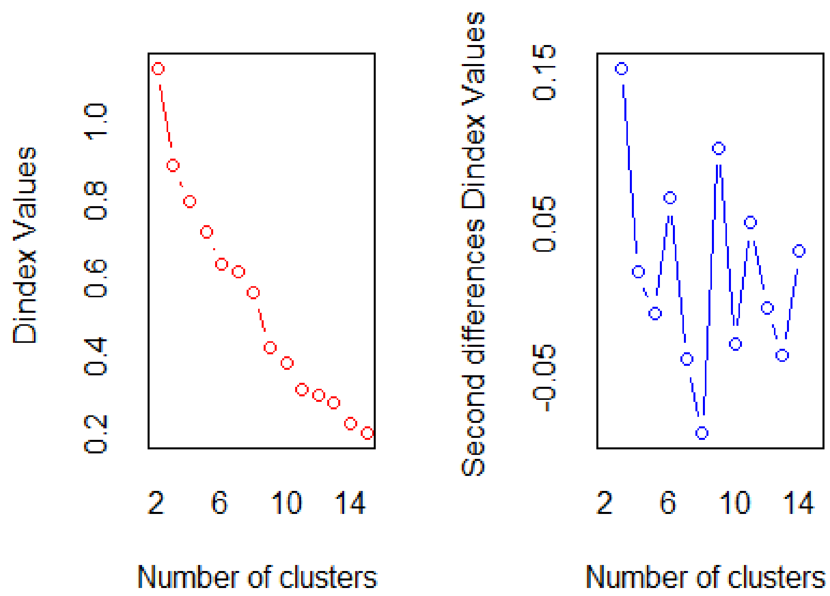
```
nb <- NbClust(br,  distance = "euclidean", min.nc=2, max.nc=15,
method = "kmeans",index = "all")
```

```
## Warning in pf(beale, pp, df2): NaNs produced

## Warning in pf(beale, pp, df2): NaNs produced
```




```
## *** : The Hubert index is a graphical method of determining the number
of clusters.
##                 In the plot of Hubert index, we seek a significant knee
that corresponds to a
##                 significant increase of the value of the measure i.e th
e significant peak in Hubert
##                 index second differences plot.
##
```



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in D index
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 4 proposed 2 as the best number of clusters
## * 5 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 2 proposed 9 as the best number of clusters
## * 3 proposed 11 as the best number of clusters
## * 3 proposed 14 as the best number of clusters
## * 4 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

The majority method says to go for 3 clusters, which seems correct

We will go with 3 clusters

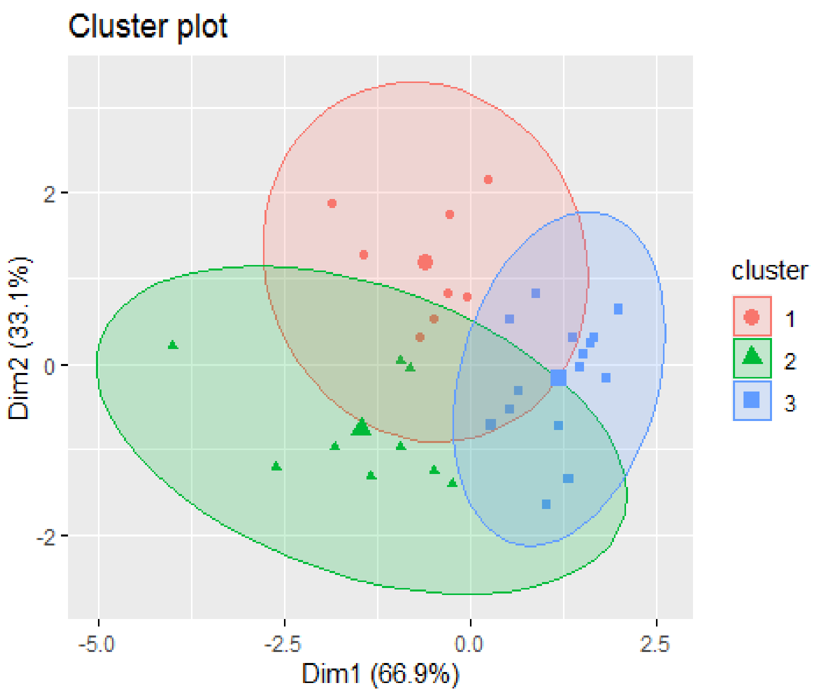
K-means clustering

3 clusters are looking good

```
km.res <- eclust(br, "kmeans", k = 3, nstart = 25, graph = FALSE)
# Visualize k-means clusters
fviz_cluster(km.res, geom = "point", frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type instead
```



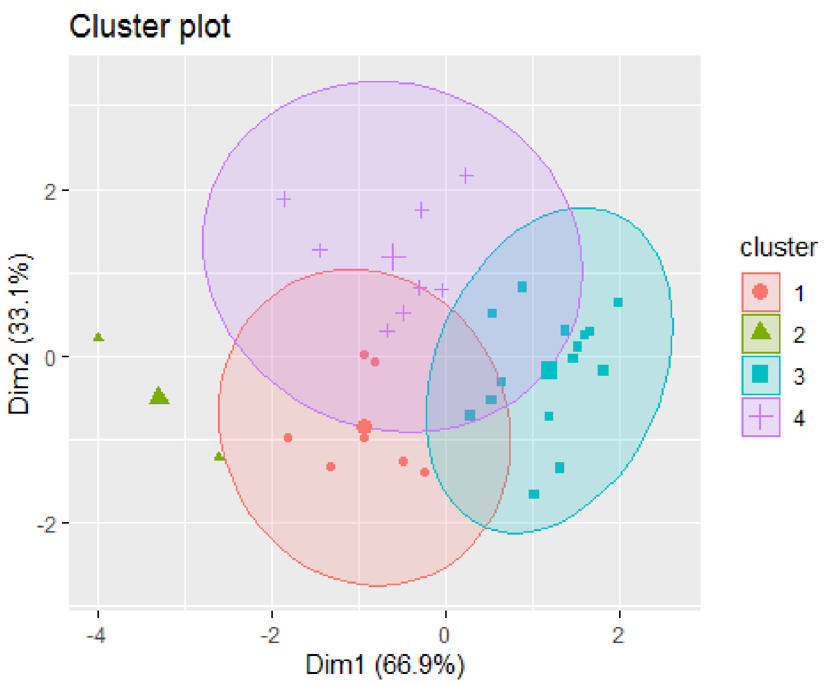
4 clusters don't look good

```
km.res <- eclust(br, "kmeans", k = 4, nstart = 25, graph = FALSE)
# Visualize k-means clusters
fviz_cluster(km.res, geom = "point", frame.type = "norm")

## Warning: argument frame is deprecated; please use ellipse instead.

## Warning: argument frame.type is deprecated; please use ellipse.type ins
tead.

## Too few points to calculate an ellipse
```

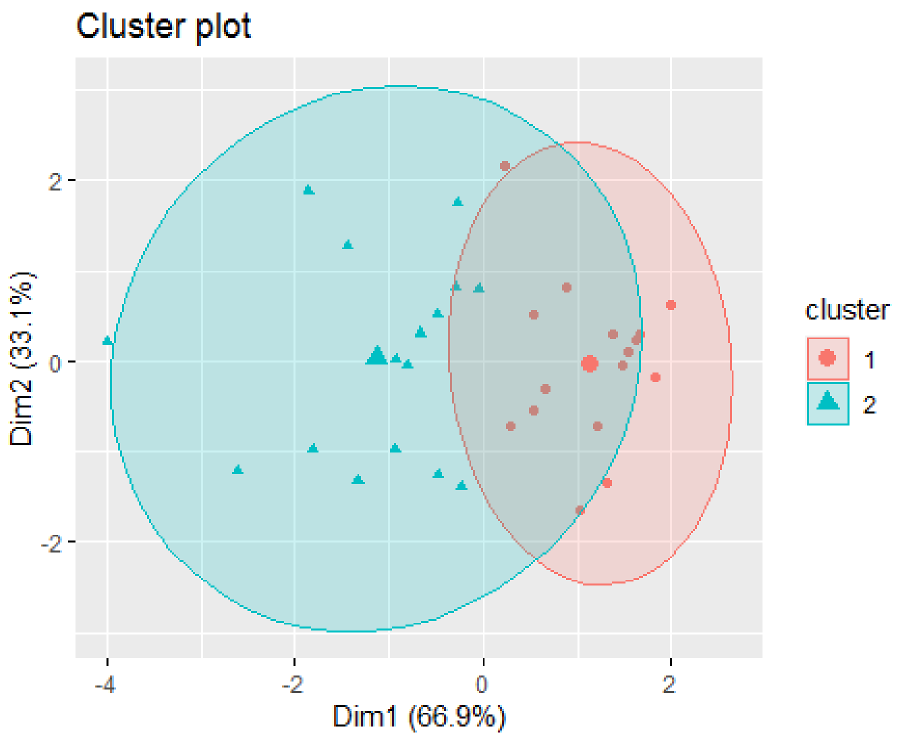


2 clusters don’t look good, either

```
km.res <- eclust(br, "kmeans", k = 2, nstart = 25, graph = FALSE)
# Visualize k-means clusters
fviz_cluster(km.res, geom = "point", frame.type = "norm")

## Warning: argument frame is deprecated; please use ellipse instead.

## Warning: argument frame.type is deprecated; please use ellipse.type ins
tead.
```



Conclusions: Here we have tried a k means clustering. cluster 3 looks good since each one of them are having similar data points. With cluster sum of squares also shows a high value.