lab5

Andreas and Priya

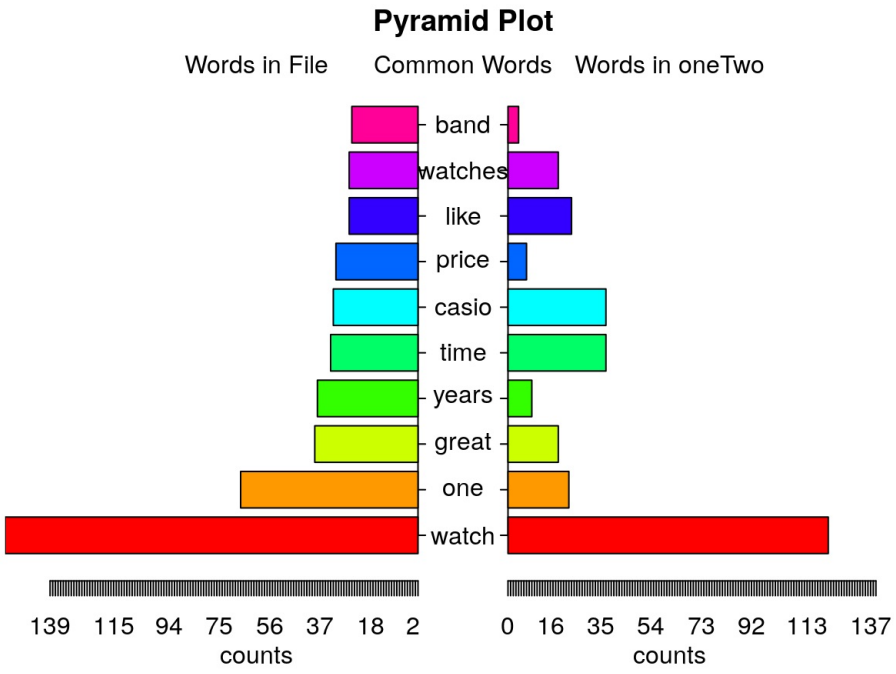
11 October 2018

##Assignment 1

###Word Cloud Plot for five

###Word Cloud Plot for One Two

###Pyramid Plot




```
## [1] 5.1 4.1 4.1 2.1
```

The word cloud for five.txt which is the good feedback has more frequent words like watch,great,one,price,casio,years,battery etc.This makes sense as words watch and casio will be often coming in the words as this is the product.As it is a positive feedback,people are happy with the price.Great is a posisive word and may be telling it last for long with word years.

THe oneTwo.txt is the negative feedback and the frequesnt words are amazon,one,back,watch,casio,battery etc.casio and watch comes frequent like before as this is the product.Amazon ,may be because they are not happy with the amazon more than product(may be the deal,return policy,shipping etc).They would like to return back the product and hence **back** has come multiple times.

###Phrasenets

Phrase nets of five.text

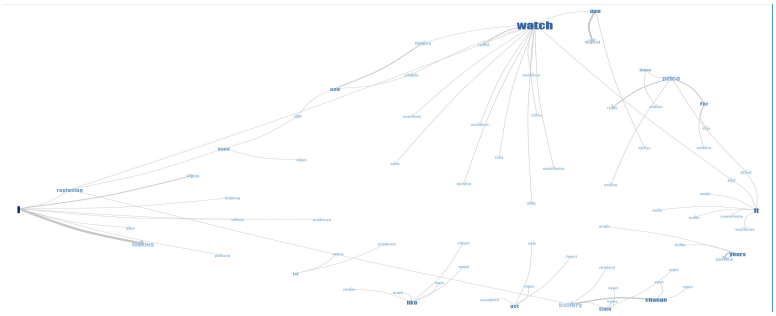


Figure 1. positive.

phrase nets of negative words

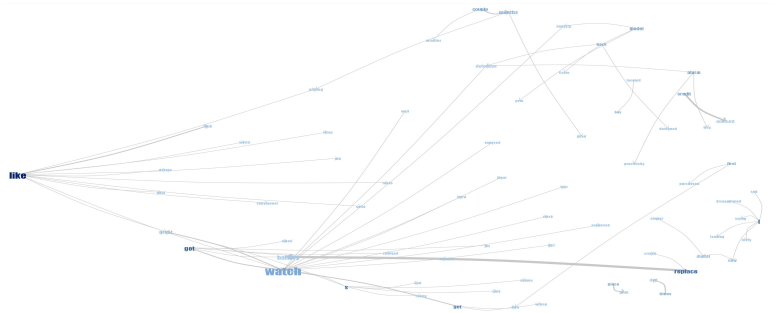


Figure 2. negative

word trees

word tree of battery with the positive feedback

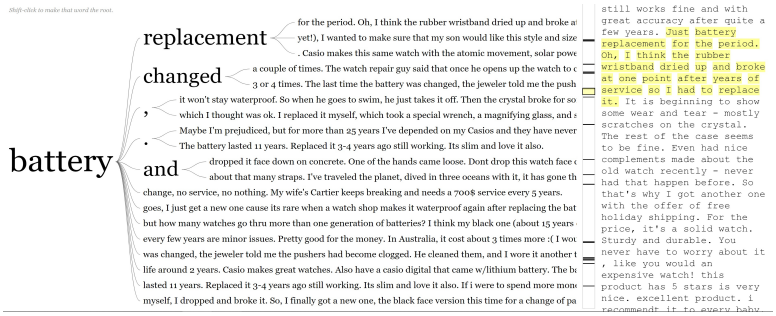


Figure 4. wordtree battery pos feedback

word tree with the word battery in negative feedback

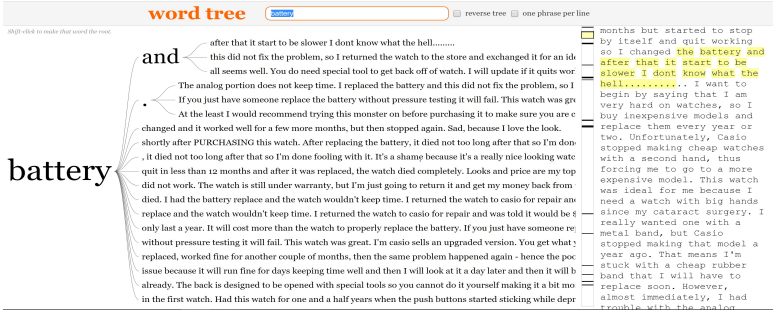


Figure 4. wordtree battery neg feedback

###Analysis

Looking at the above graphs we come to conclusions like

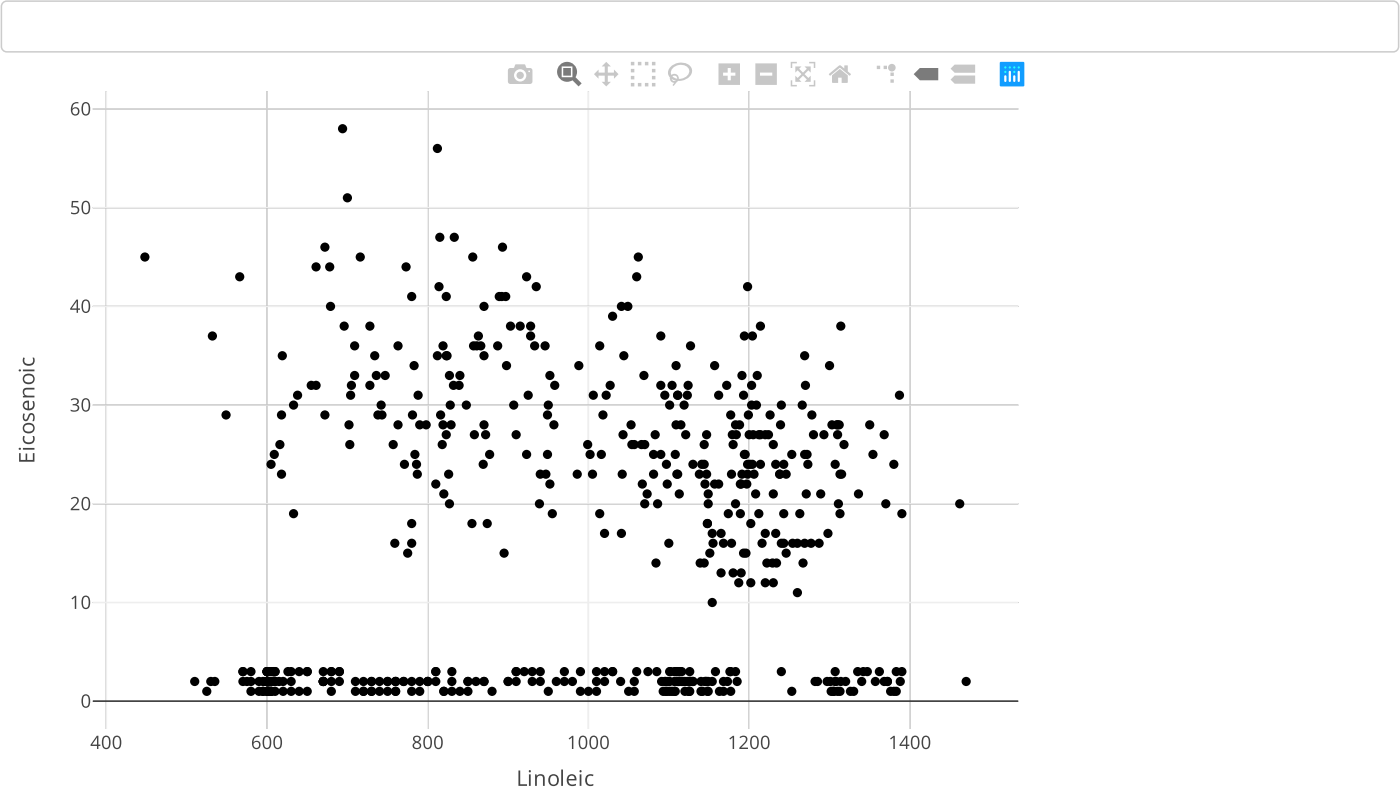
THe watch has analog and dgital display and it appears in black,gold and white.It has sporty look,rubbery band and waterprrof and good value for money.The satisfied customers are talking about modern look,water resistance,some of them say the battery last long and it is good priced.Dis statisfied customers talk about the defective alrams,battery once changed the watch does not work well,The analog portions becomes defective etc.Also Replacing battery is costly .

Seems like the customers are satisfied with the look and feel and price of the watch .With some changesin the battery and all it can be a good product.

##Assignment 2

###1.Scatter Plot eicoseinoic vs linoleic

Choose region



The value of eicosenoic is between one and three. ###2.Linked Scatterplot and Bar chart

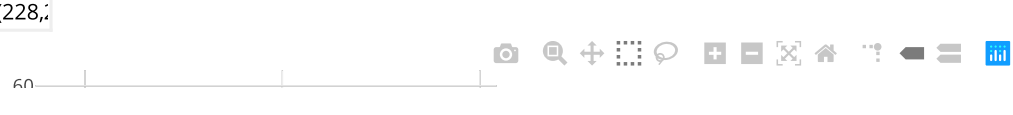
values of stearic

52 375

152 248 344

Brush color

rgba(228,



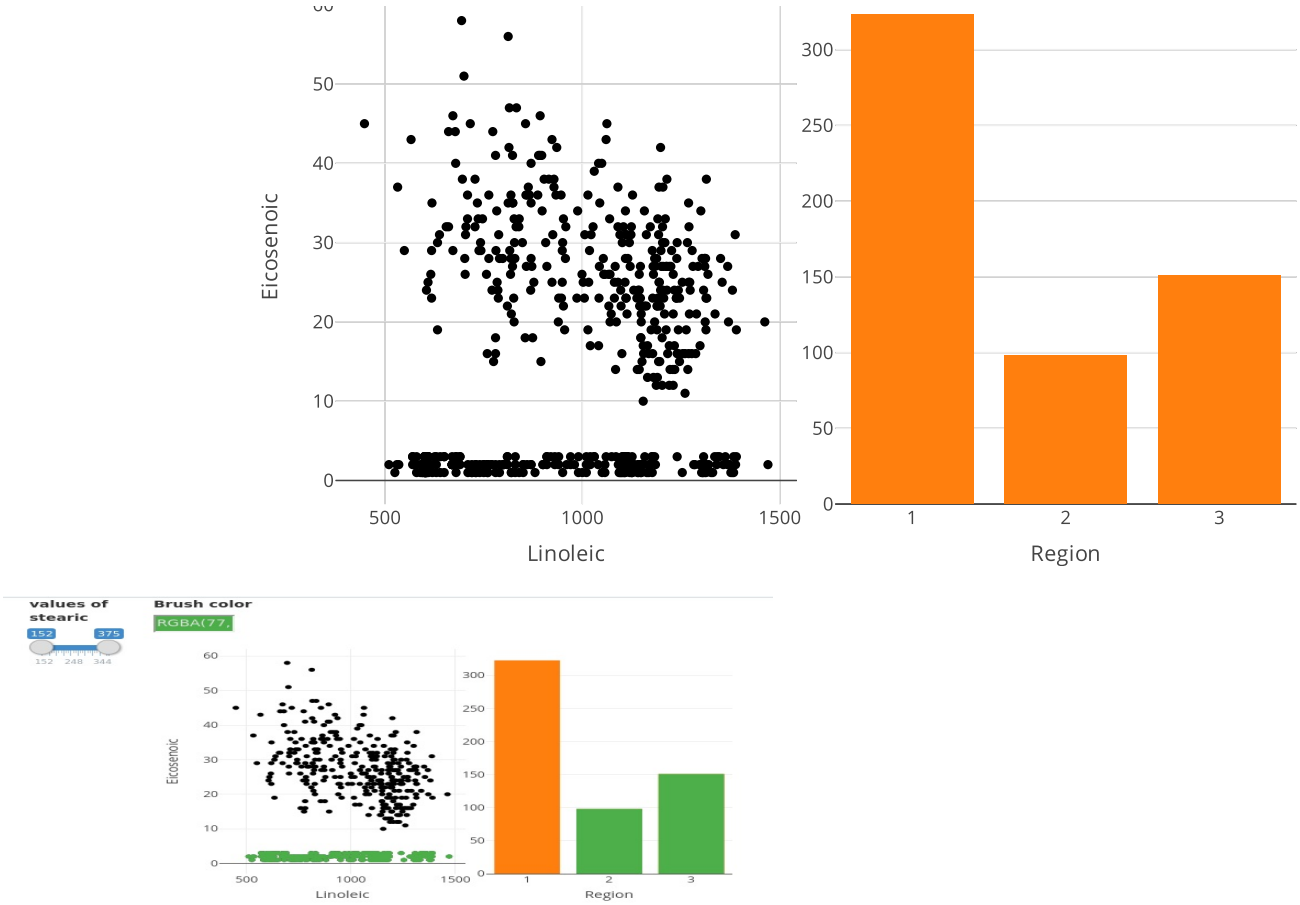


Figure 1. coloured

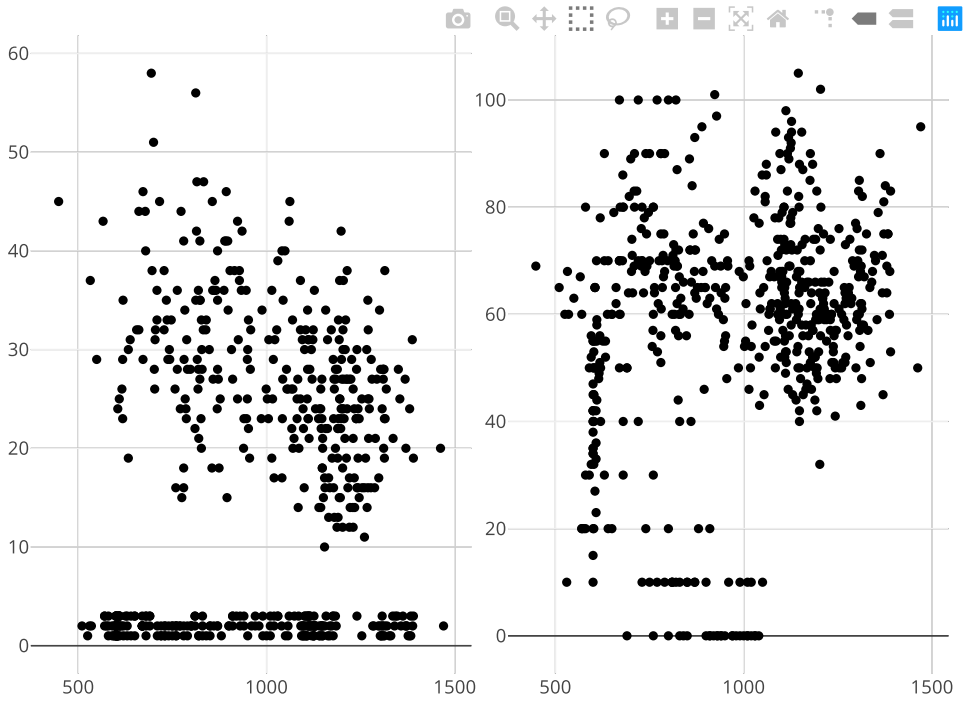
Using brushing when we take out the lower values of eicosenoic(between 1 and 3),we can see they correspond to region south and sardinia Island.In south region the amount of stearic acid is between 190 and 270.

The operators used here are seelction operators for brushing,Connection operator for connecting the plots and Filtering operator for filtering stearic values

###3.Linked Scatter Plots

Brush color

rgba(228,2



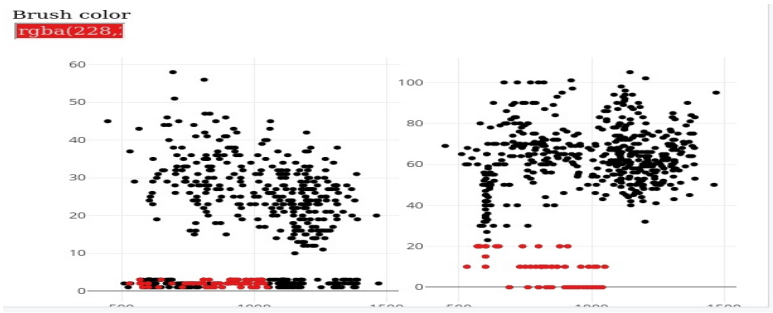


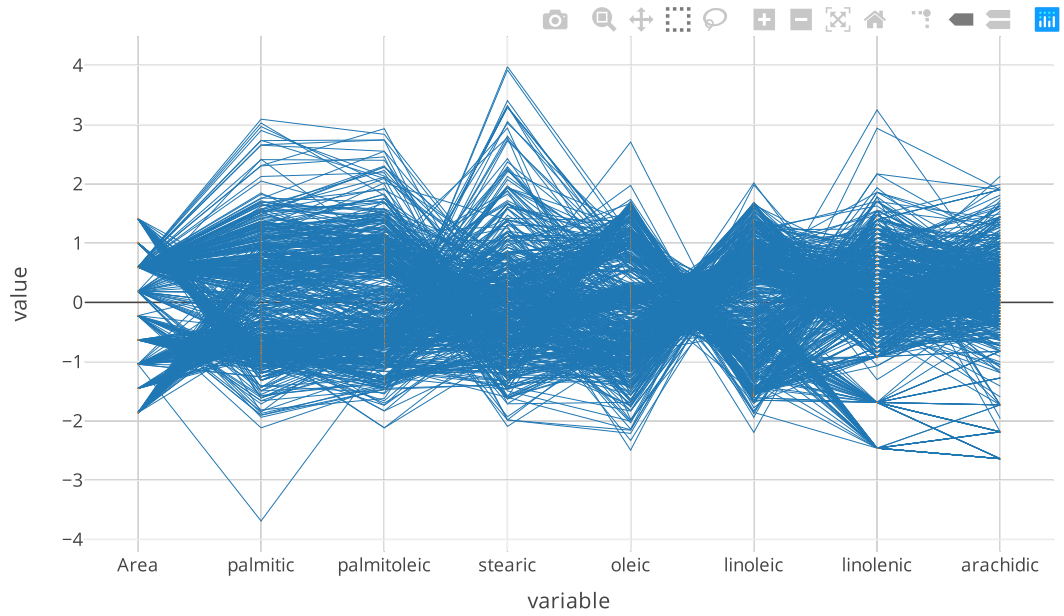
Figure 1. linked scatter

The values of arachidic below 40 are outliers in the plot2 are also oultiers in plot 1.IN plot one their eicosenoic values are between 1 and 3.

###4.Parallel Coordinate Plot linked Barplot and Scatterplot

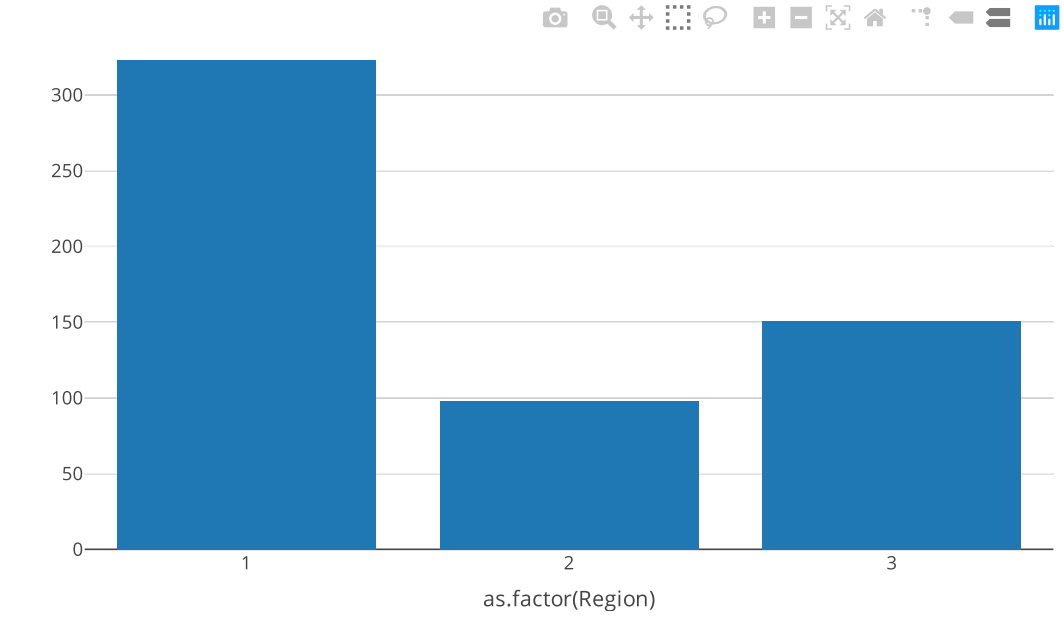
Brush color

rgba(228,2



Brush color

rgba(228,2



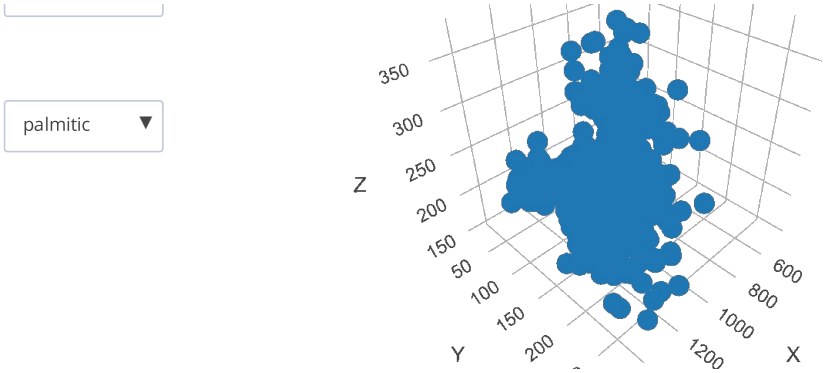
Brush color

rgba(228,2

palmitic

Distribution of Oil

palmitic



When we analysed the parallel coordinate plot, linolenic, oleic and palmitonic seemed to have more clusters hence they are influentional variables to differentiate regions.

In South region we can observe multiple clusters. linolenic, oleic and linoleic forms cluster, steraric, oleic, linoleic forms another cluster.

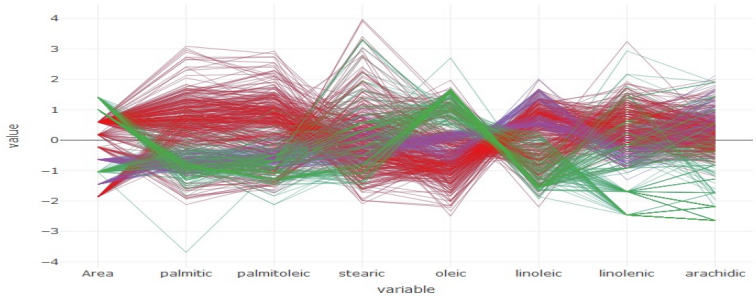


Figure 1. plot_123

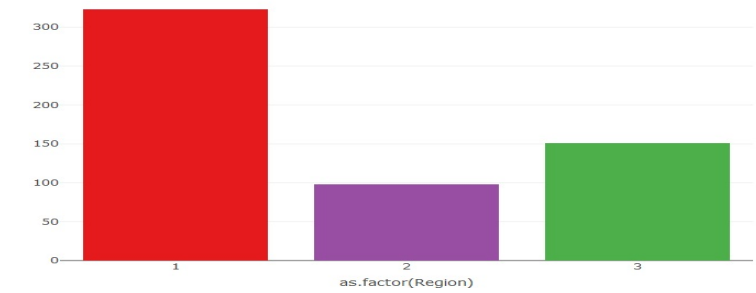


Figure 1. plot_123

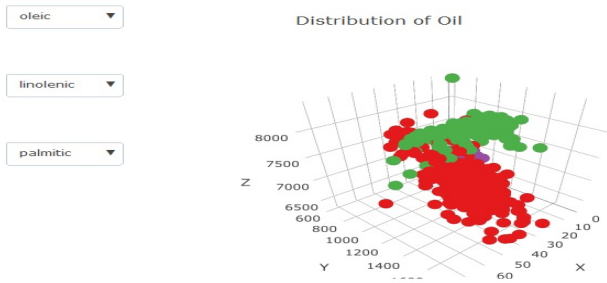


Figure 1. plot_123

As seen in the above screenshots, when we select the 3 variables and brush the regions with 3 different colours, the plot is almost clusterd in 3 seperate groups.

###5.Analysis

In step 4 we have used:

selection operator while brushing the plots ,connection operator to connect plots, Reconfiguring operator dynamically selectiong axix variables in 3D plot on data value space operand

An additional interaction operator which can be used is Filtering operator.This operator can be used to filter based on regions and analyse the concentration of acids in oil from each region. This data can be used for analysing the region of origin of an oil given the concentration of acids in that oil.s

##Appendix

```
## ----message=FALSE,echo=FALSE----------------------------------------------
library(tm)
library(wordcloud)
```

```
library(RColorBrewer)
#library(viridisLite)
library(plotrix)
library(tidyverse)


## ----message=FALSE,echo=F-------------------------------------------------
five<-read.table("Five.txt",header=F, sep='\n',encoding="latin1")#Read file
five$doc_id=1:nrow(five)
colnames(five)[1]<-"text"
mycorpus <- Corpus(DataframeSource(five)) #Creating corpus (collection of text data)

clean_corpus <- function(corpus){
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus,function(x) removeWords(x,stopwords("english")))
  return(corpus)
}


mycorpus <- clean_corpus(mycorpus)
tdm <- TermDocumentMatrix(mycorpus) #Creating term-document matrix
m <- as.matrix(tdm)

#here we merge all rows
v <- sort(rowSums(m),decreasing=TRUE) #Sum up the frequencies of each word
d <- data.frame(word = names(v),freq=v) #Create one column=names, second=frequences

pal<-brewer.pal(6,"PRGn") #Create palette of colors
#color_pal <- cividis(n = 8)


#wordcloud(d$word,d$freq, scale=c(8,.3),min.freq=2,max.words=100, random.order=F, rot.per=.15, colors=col_pal, vfo
nt=c("sans #serif","plain"))

wordcloud(d$word,d$freq, scale=c(8,.3),min.freq=2,max.words=100, random.order=F, rot.per=.15, colors=pal, vfont=c(
"sans serif","plain"))



## ----message=F,echo=F-----------------------------------------------------
oneTwo<-read.table("OneTwo.txt",header=F,sep="\n")
oneTwo$doc_id=1:nrow(oneTwo)
colnames(oneTwo)[1]<-"text"

mycor <- Corpus(DataframeSource(oneTwo))

mycor <- clean_corpus(mycor)
tdm1<- TermDocumentMatrix(mycor) #Creating term-document matrix
m1 <- as.matrix(tdm1)

#here we merge all rows
v1 <- sort(rowSums(m1),decreasing=TRUE) #Sum up the frequencies of each word
d1<- data.frame(word = names(v1),freq=v1)
pal<-brewer.pal(8,"Dark2")

wordcloud(d1$word,d1$freq, scale=c(8,.3),min.freq=2,max.words=100, random.order=F, rot.per=.15, colors=pal, vfont=
c("sans serif","plain"))


## ----message=FALSE,warning=FALSE,echo=F-----------------------------------
j<-left_join(d,d1,by="word")

pyramid.plot(
  j$freq.x[1:10],j$freq.y[1:10],
  # Words
  labels = j$word[1:10],
  top.labels = c("Words in File", "Common Words", "Words in oneTwo"),
  main = "Pyramid Plot",gap=17,unit="counts")



## ----message=FALSE,echo=FALSE---------------------------------------------
library(tidyverse)
library(plotly)
library(plyr)
library(crosstalk)
library(GGally)
```

```
olive<-read.csv("olive.csv")
olive$Region<-as.factor(olive$Region)
levels(olive$Region)<-c("1","2","3")
o<-SharedData$new(olive)
oo<-SharedData$new(olive,~Region,group = "Choose region")


## ----set-options,message=F,echo=F-----------------------------------------

scatterolive <- plot_ly(oo, y = ~eicosenoic, x = ~ linoleic)%>%group_by(Region)%>%
  add_markers(color = I("black"),name="hollow")%>%highlight(on="plotly_hover",persistent = F,selectize = T) %>%  l
ayout(xaxis = list(title = "Linoleic"),
      yaxis = list(title = "Eicosenoic"))

scatterolive


## ----message=FALSE,warning=FALSE,echo=F-----------------------------------



barolive <- plot_ly(oo, x=~as.factor(Region))%>%add_histogram()%>%layout(barmode="overlay",xaxis=list(title="Regio
n"))

bscols(widths=c(2, NA),filter_slider("stearic", "values of stearic", oo, ~stearic),subplot(scatterolive,barolive,t
itleY = TRUE,  titleX = TRUE)%>%highlight(on="plotly_select", dynamic=T, persistent = T, opacityDim = I(1))%>%hide
_legend())





## ----message=FALSE,echo=F-------------------------------------------------

scatter1 <- plot_ly(o, x = ~linoleic, y = ~eicosenoic) %>%
  add_markers(color = I("black"))  %>% layout(xaxis = list(title = "linoleic"),
                                      yaxis = list(title = "Eicosenoic"))
scatter2 <- plot_ly(o, x = ~linoleic, y = ~arachidic) %>%
  add_markers(color = I("black"))%>% layout(xaxis = list(title = "linoleic"),
                                         yaxis = list(title = "arachidic"))


subplot(scatter1,scatter2)%>%
  highlight(on="plotly_select", dynamic=T, persistent=T, opacityDim = I(1))%>%hide_legend()




## ----message=FALSE,fig.width=20, fig.height=18,echo=F---------------------
oliveparralelcord<-ggparcoord(olive, columns = c(3:10))


d<-plotly_data(ggplotly(oliveparralelcord))%>%group_by(.ID)
d1<-SharedData$new(d, ~.ID, group="olive")

p1<-plot_ly(d1, x=~variable, y=~value)%>%
  add_lines(line=list(width=0.3))%>%
  add_markers(marker=list(size=0.3),
              text=~.ID, hoverinfo="text")


oildata <- olive
oildata$.ID=1:nrow(oildata)
d2<-SharedData$new(oildata, ~.ID, group="olive")
p2 <- plot_ly(d2, x=~as.factor(Region))%>%add_histogram()%>%layout(barmode="overlay")


ButtonsX=list()
for (i in 4:10){
  ButtonsX[[i-3]]= list(method = "restyle",
                        args = list( "x", list(olive[[i]])),
                        label = colnames(olive)[i])
}


ButtonsY=list()
for (i in 4:10){
  ButtonsY[[i-3]]= list(method = "restyle",
```

```
                        args = list( "y", list(olive[[i]])),
                        label = colnames(olive)[i])
}

ButtonsZ=list()
for (i in 4:10){
  ButtonsZ[[i-3]]= list(method = "restyle",
                        args = list( "z", list(olive[[i]])),
                        label = colnames(olive)[i])
}


axx <- list(
  title = "X"
)

axy <- list(
  title = "Y"
)

axz <- list(
  title = "Z"
)
 p3 <- plot_ly(d2,x= ~linoleic,y = ~palmitoleic, z = ~stearic)%>%add_markers()%>%
  layout(xaxis=list(title=""), yaxis=list(title=""),
         title = "Distribution of Oil",
         updatemenus = list(
           list(y=0.9, buttons = ButtonsX),
           list(y=0.6, buttons = ButtonsY),
           list(y=1.2, buttons = ButtonsZ)
         )  )%>%layout(scene = list(xaxis=axx,yaxis=axy,zaxis=axz))




ps<-htmltools::tagList(p1%>%
                         highlight(on="plotly_select", dynamic=T, persistent = T, opacityDim = I(1))%>%
                         hide_legend(),
                       p2%>%
                         highlight(on="plotly_select", dynamic=T, persistent = T, opacityDim = I(1))%>%
                         hide_legend(),
                       p3%>%
                         highlight(on="plotly_click", dynamic=T, persistent = T, opacityDim = I(1))%>%
                         hide_legend()
)
htmltools::browsable(ps)
```