Lab2

Priya(priku577) and Andreas(andch552)

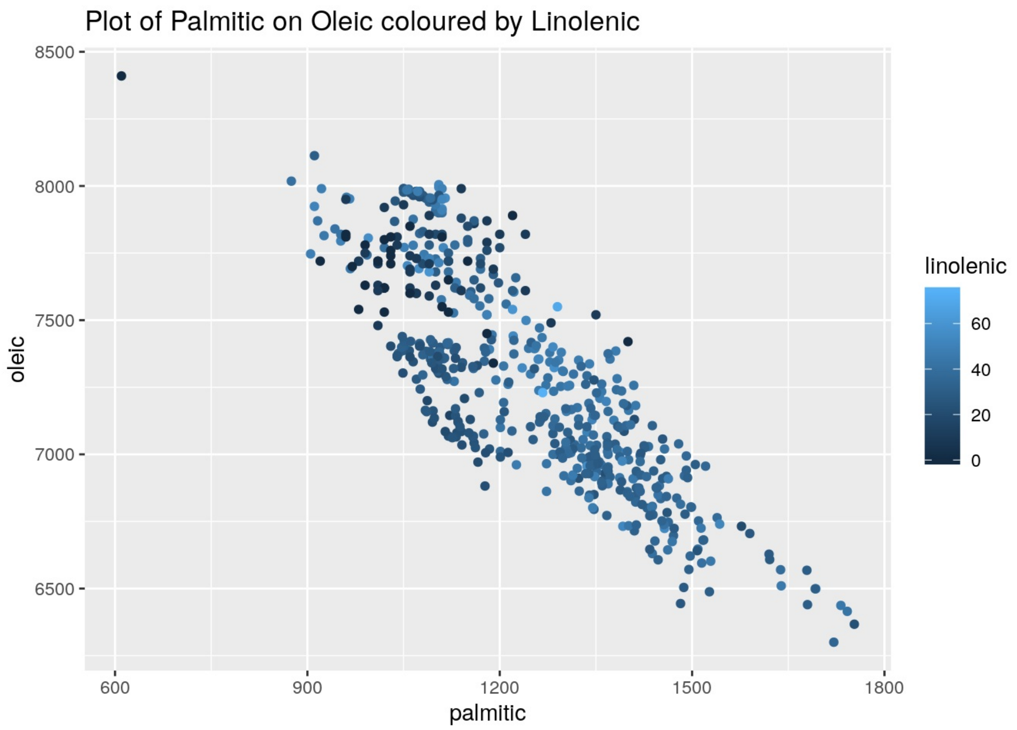
19 September 2018

###Assignment 1

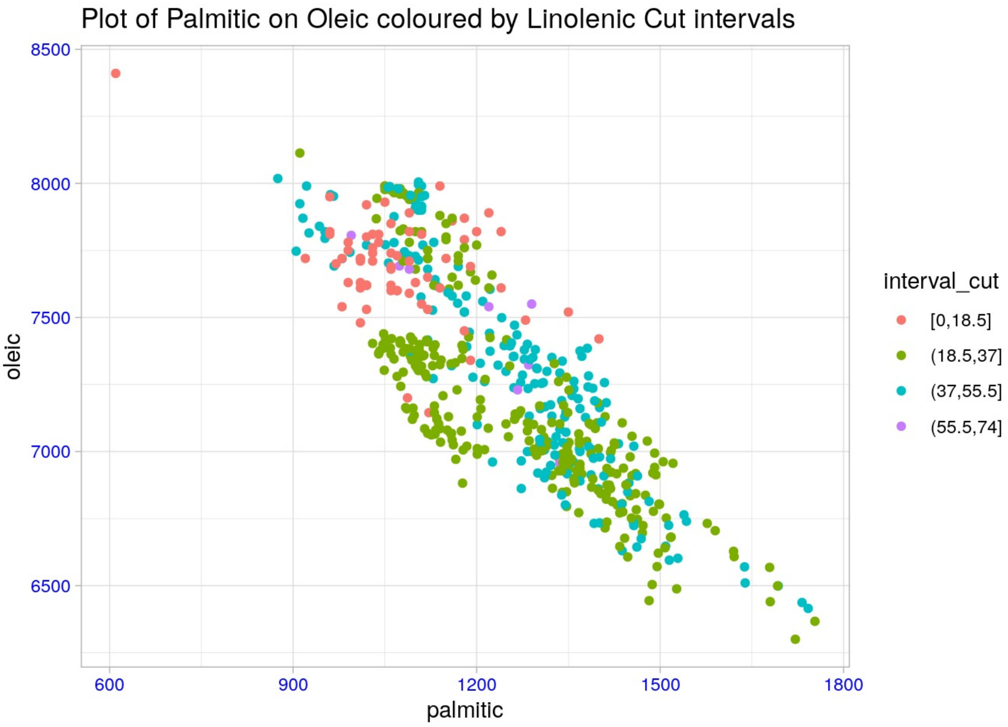
##1

scatterplot in Ggplot2 that shows dependence of Palmitic on Oleic in which observations are colored by Linolenic

```
library(ggplot2)
library(plotly)
library(MASS)

olive<-read.csv("olive.csv",header=TRUE, sep=",")

p1<-ggplot(olive,aes(palmitic,oleic,col=linolenic))+geom_point()+labs(x="palmitic",y="oleic")+ggtitle("Plot of Pal
mitic on Oleic coloured by Linolenic")
p1
```



```
interval_cut <- cut_interval(olive$linolenic, 4)
p2<-ggplot(olive,aes(palmitic,oleic,col=interval_cut))+geom_point()+theme_light()+labs(x="palmitic",y="oleic")+ggt
itle("Plot of Palmitic on Oleic coloured by Linolenic Cut intervals")+theme(axis.text = element_text(colour = "blu
e"))
p2
```



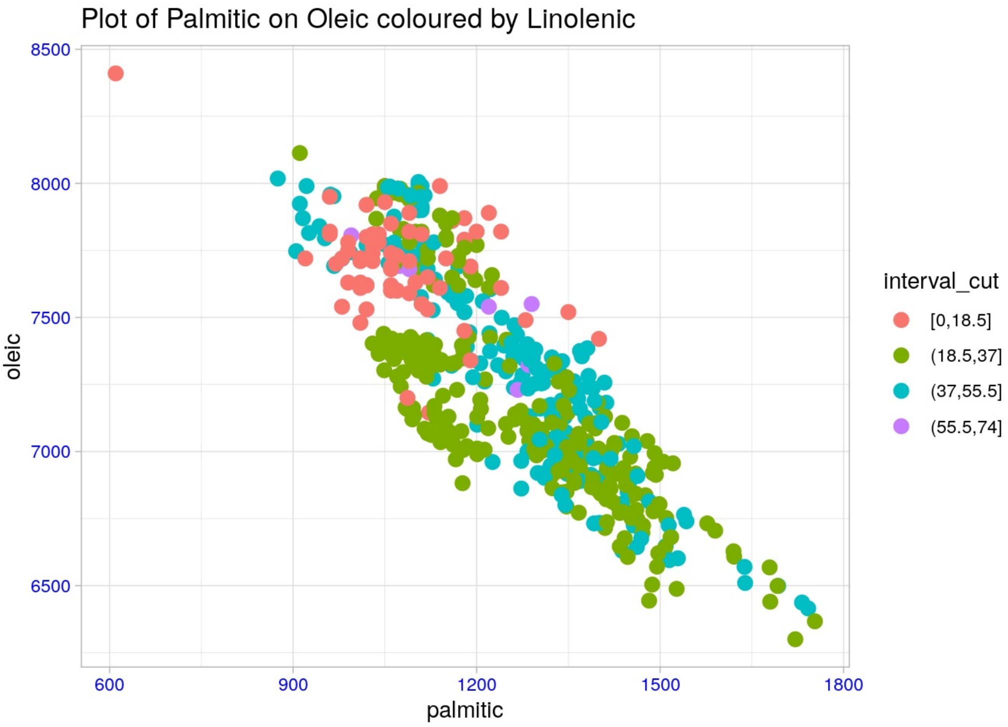
Channel capacity of the human perception is the perception problem told here.The perception consists of three steps: recognizing, organizing and interpreting.Grouping is hard to do in the first plot as different shades of same hue are used. THe groups can be easily identified by perception in the second plot.Its difficult to distinguish between different shades of same hue while its easier to distinguish between differnet hues (channel capacity is here).

##2

scatterplots of Palmitic vs Oleic in which you map the discretized Linolenic with four classes to:

a. Color

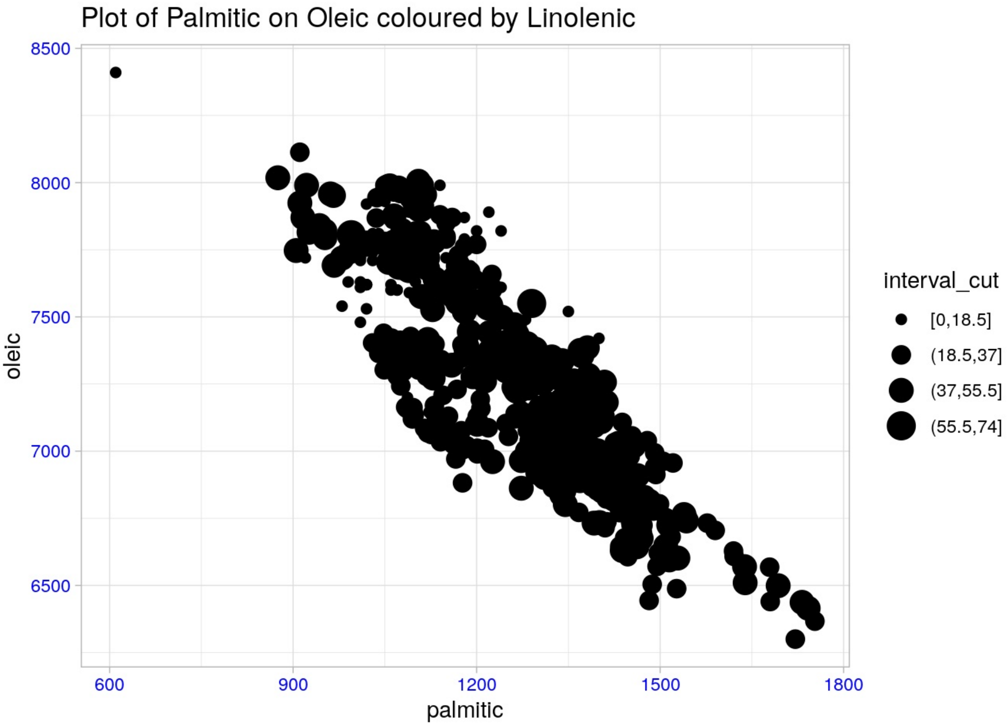
```
p3<-s2<-ggplot(olive,aes(palmitic,oleic,col=interval_cut))+geom_point()+theme_light()+labs(x="palmitic",y="oleic")
+geom_point(size=3)+ggtitle("Plot of Palmitic on Oleic coloured by Linolenic")+theme(axis.text = element_text(colo
ur = "blue"))
p3
```



b. Size

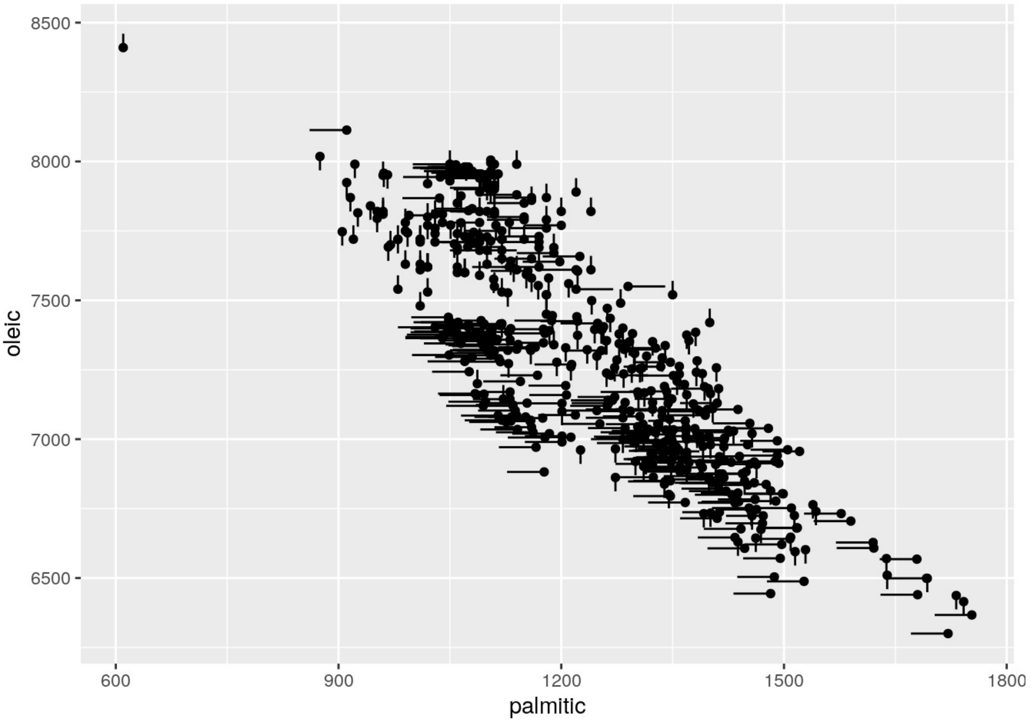
```
p4<-s2<-ggplot(olive,aes(palmitic,oleic))+geom_point(aes(size=interval_cut))+theme_light()+labs(x="palmitic",y="ol
eic")+geom_point()+ggtitle("Plot of Palmitic on Oleic coloured by Linolenic")+theme(axis.text = element_text(colou
r = "blue"))
p4
```

```
## Warning: Using size for a discrete variable is not advised.
```



c. Orientation angle

```
angleplot<-ggplot(olive,aes(x=palmitic,y=oleic)) +
  geom_point() +geom_spoke(aes(angle = as.numeric(cut_interval(linolenic, 4))*pi/2), radius = 50)
angleplot
```

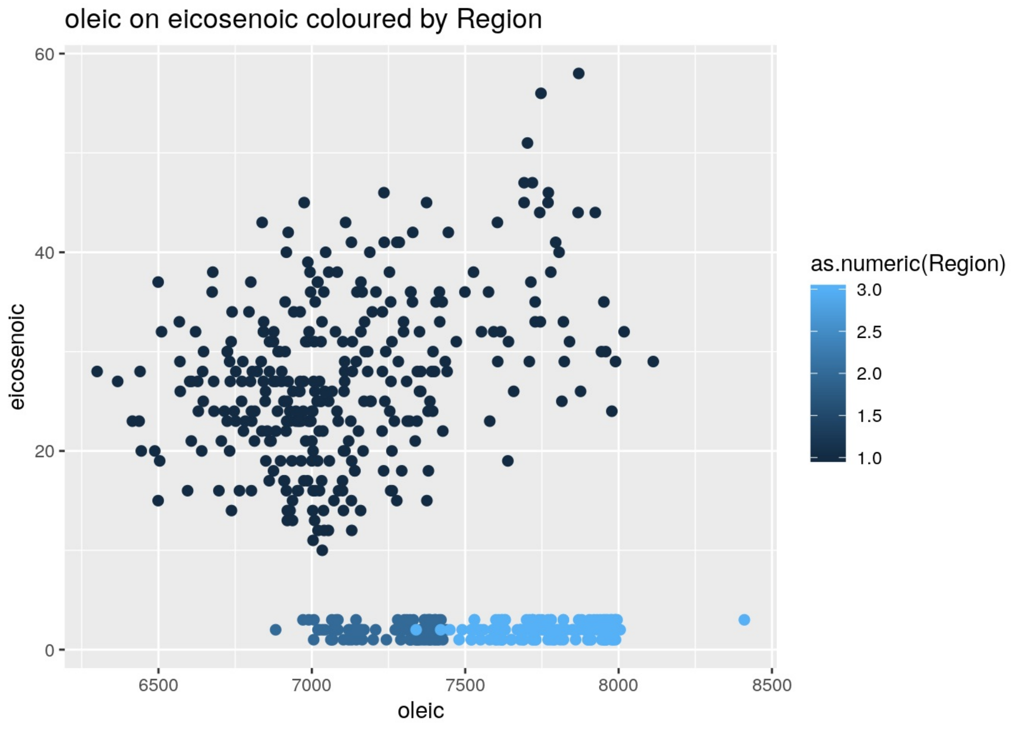


We feel it is very difficult to differenciate between categories in the second plot with size (b.size).The third plot is not good either.The first graph is obviously the best. The coloured graph is 3.1 bits the size plot 2.2 bits and the angle plot is 2.8 bits

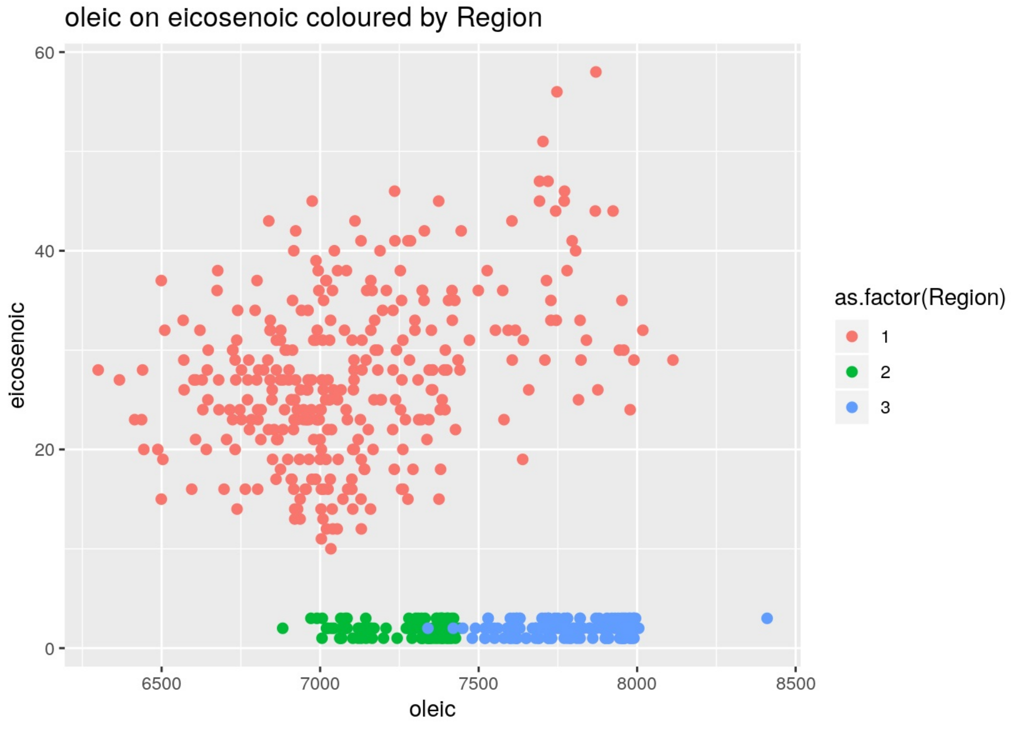
##3

Create a scatterplot of Oleic vs Eicosenoic in which color is defined by numeric values of Region.

```
p6<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=as.numeric(Region)))+labs(x="oleic",y="eicosenoic")+geom_point(siz
e=2)+ggtitle("oleic on eicosenoic coloured by Region")
p6
```



```
p7<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=as.factor(Region)))+labs(x="oleic",y="eicosenoic")+geom_point(size
=2)+ggtitle("oleic on eicosenoic coloured by Region")
p7
```



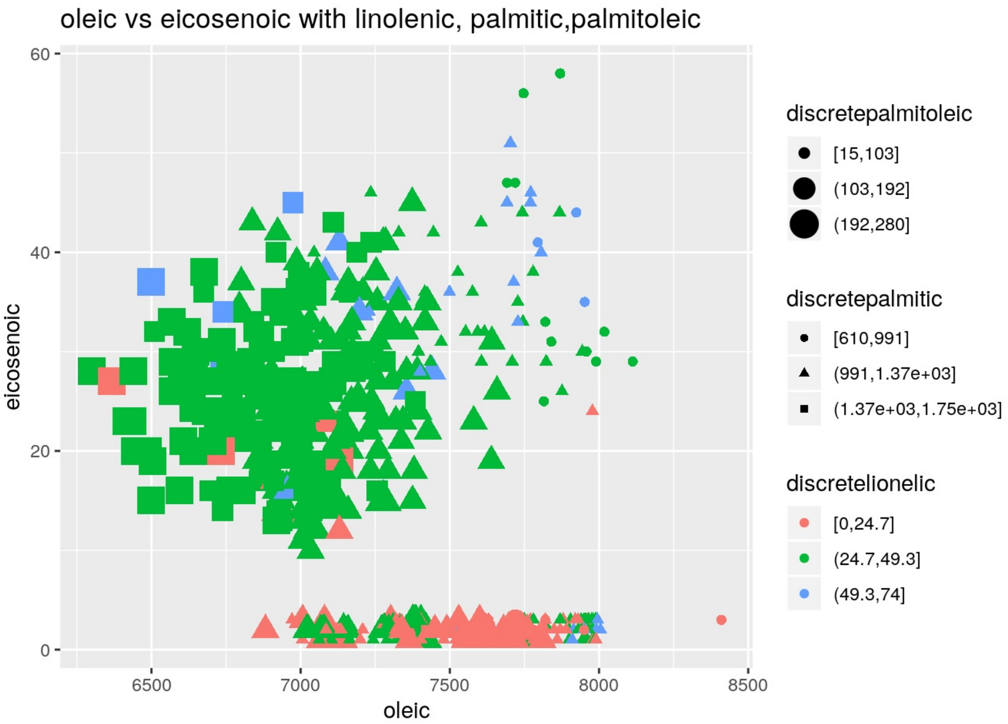
In the second plot it is very easy to find the decision boundaries as using discrete hues is a preattendive feature and identifying the boundaries take very less time.

##4

scatterplot of Oleic vs Eicosenoic in which color is defined by a discretized Linoleic (3 classes), shape is defined by a discretized Palmitic (3 classes) and size is defined by a discretized Palmitoleic (3 classes)

```
discretelionelic<-cut_interval(olive$linolenic, 3)
discretepalmitic<-cut_interval(olive$palmitic, 3)
discretepalmitoleic<-cut_interval(olive$palmitoleic, 3)

p8<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=discretelionelic,shape=discretepalmitic,size=discretepalmitoleic))
+geom_point()+ggtitle("oleic vs eicosenoic with linolenic, palmitic,palmitoleic")
p8
```

```
## Warning: Using size for a discrete variable is not advised.
```



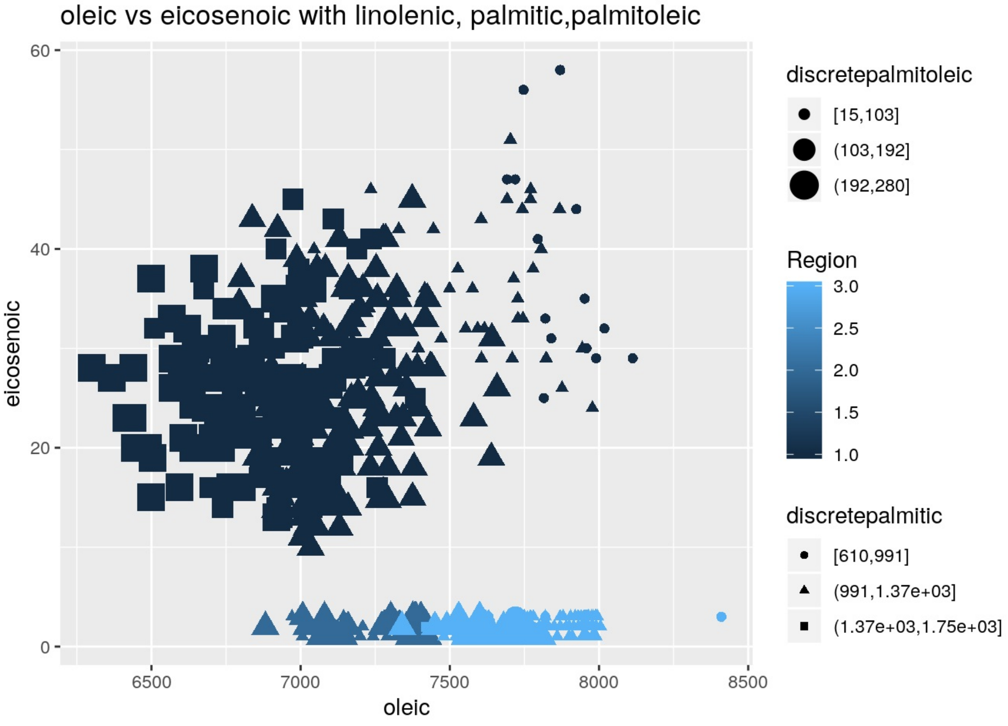
It is really difficult to differentiate between 27 different types of observations as there are no clear cut boundaries. Here there are three shades of colour and three types of shapes.Their summation is not the same as summation of the channel capacity of three shapes and three colours.

##5

scatterplot of Oleic vs Eicosenoic in which color is defined by Region,shape is defined by a discretized Palmitic (3 classes) and size is defined by a discretized Palmitoleic (3 classes)

```
p9<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=Region,shape=discretepalmitic,size=discretepalmitoleic))+geom_poin
t()+ggtitle("oleic vs eicosenoic with linolenic, palmitic,palmitoleic")
p9
```

```
## Warning: Using size for a discrete variable is not advised.
```

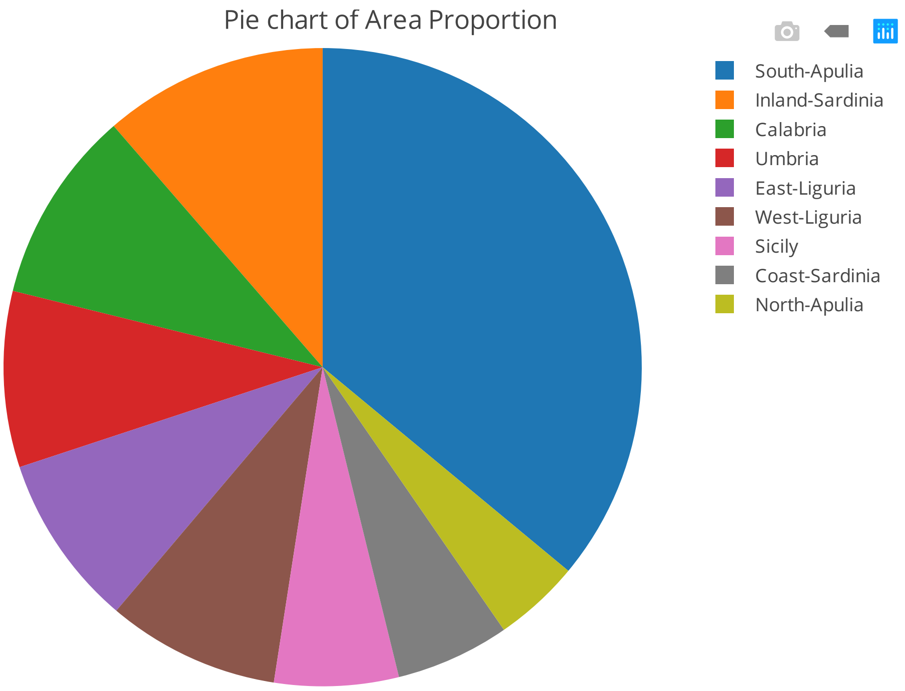


The boundary between the regions is easily identified as we have colour to distinguish the boundaries.Colour,orientation and intensity can be automatically percieved by human eye according to treistman theory of integration and the combination of thes may require time to distinguish as seen in the quetion 4.Here colour is the one that helps as distinguish faster.

##6

create a pie chart that shows the proportions of oils coming from different Areas using plotly

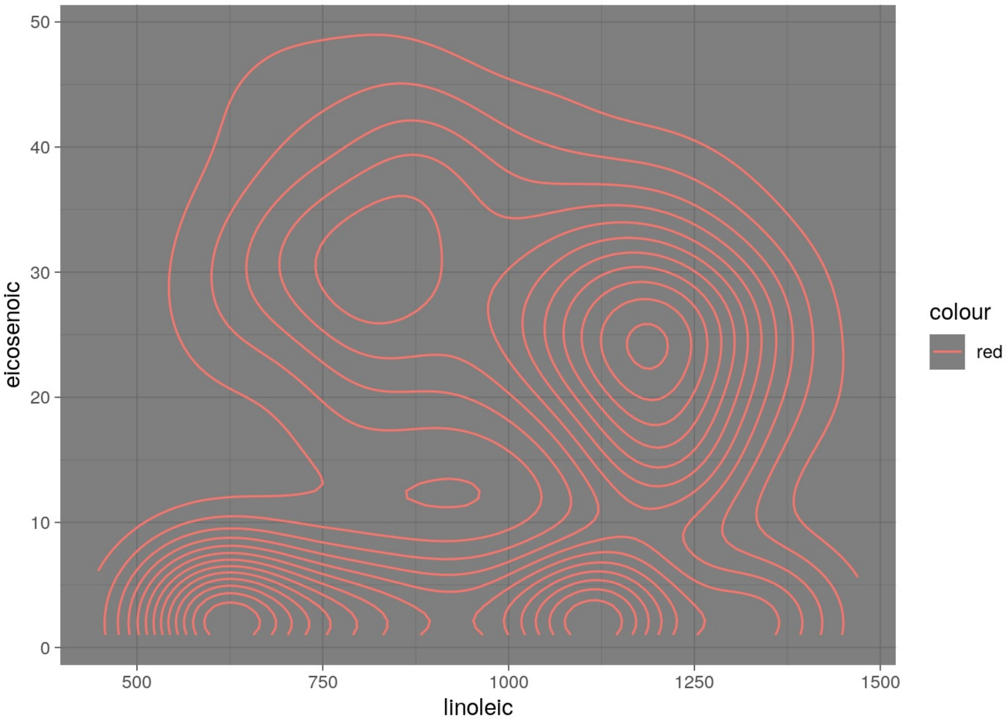
```
library(plotrix)
library(plotly)

mytable <- table(olive$Area)
dframe<-as.data.frame(mytable)

#pie3D(dframe$Freq,labels=dframe$Var1,explode=0.1,main="Pie Chart of Area Proportion ")

p10 <- plot_ly(dframe, labels = ~Var1, values = ~Freq, type = 'pie',textinfo = "none", hoverinfo = "text",text=~Va
r1) %>%
  layout(title = 'Pie chart of Area Proportion',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

p10
```



THe pie chart is less effective than the bar chart.When the values differ by small number it is diffcult to predict which area’s frequency was more without hovering and finding for the exact value.In barchart just looking at the lenght we can find this. ##7

2d-density contour plot

```
p11<-ggplot(olive, aes(x=linoleic, y=eicosenoic,col="red") ) +geom_density_2d(position="identity")+theme_dark()
p11
```



Outliers are missed in contour plots.

###Assignment 2

Load the baseball data in r

```
library(readxl)
library(ggplot2)
library(plotly)
library(MASS)


df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)
```

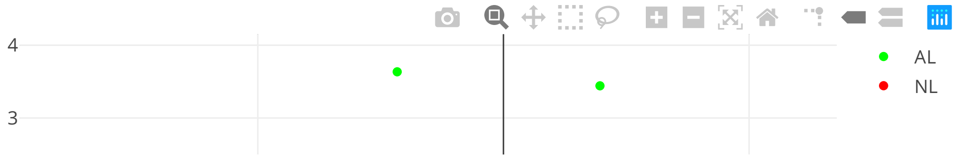
It is reasonable to scale the data because we have different range of features in the data.

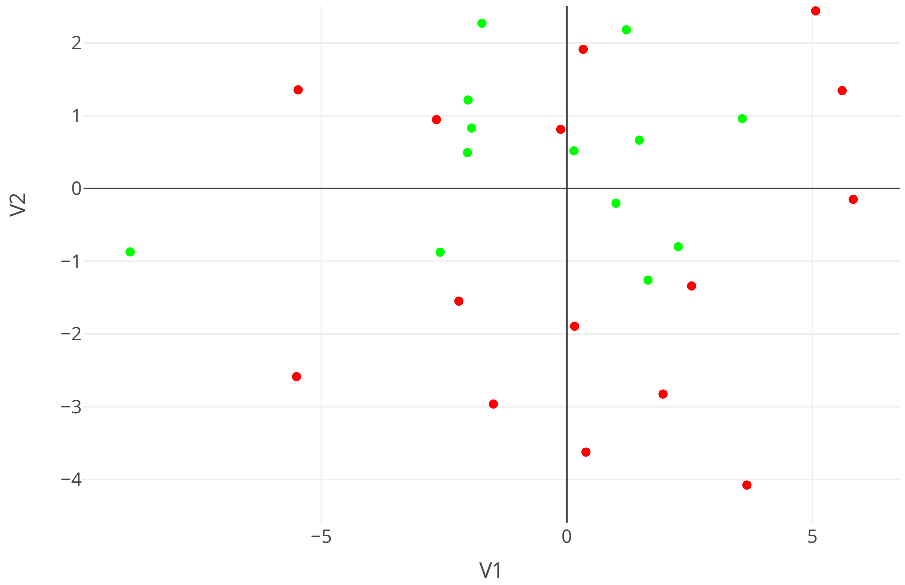
```
library(readxl)

df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)
df<-data.frame(df)

scaled_df<-scale(df[,-c(1,2)])
d<-dist(scaled_df)
mds<-isoMDS(d,k=2)
```

```
## initial  value 19.856833
## iter   5 value 16.319153
## iter  10 value 16.046215
## final  value 15.935476
## converged
```

```
dt<-(mds$points)
dt_new<-as.data.frame(dt)
dt_new$League<-df[,2]
dt_new$Team<-df[,1]



plot_ly(dt_new, x=~V1, y=~V2, type="scatter", hovertext=~Team, color=~League, colors=c("green", "red"))
```

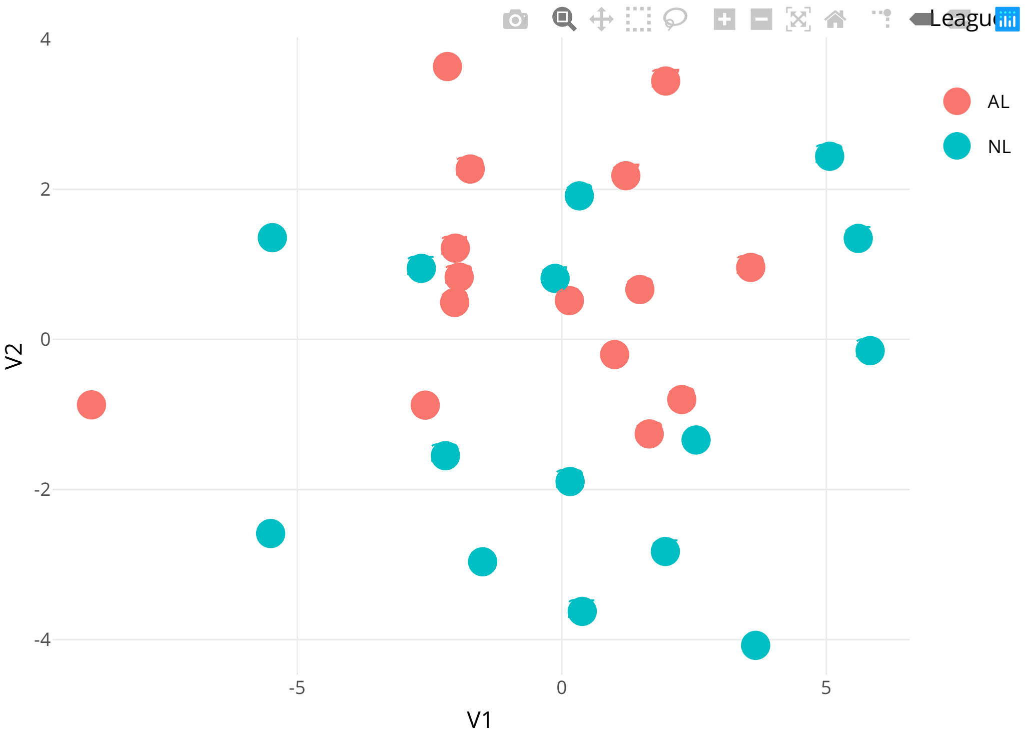
```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
```





Form the graph we can first observe that Boston Red Sox,Baltimore orles,philadelphia philles seems to be outliers.Theres seems to be some kind of diffrence between the Leagues.V2 seems to provide a better differenciation between leagues.ALso the NL is more scattered than AL.

```
p <- ggplot(data=dt_new,aes(x=V1,y=V2,col=League,size=5))+geom_point()+theme_minimal()+
  geom_text(aes(label=rownames(dt_new),size=5))
ggplotly(p)
```

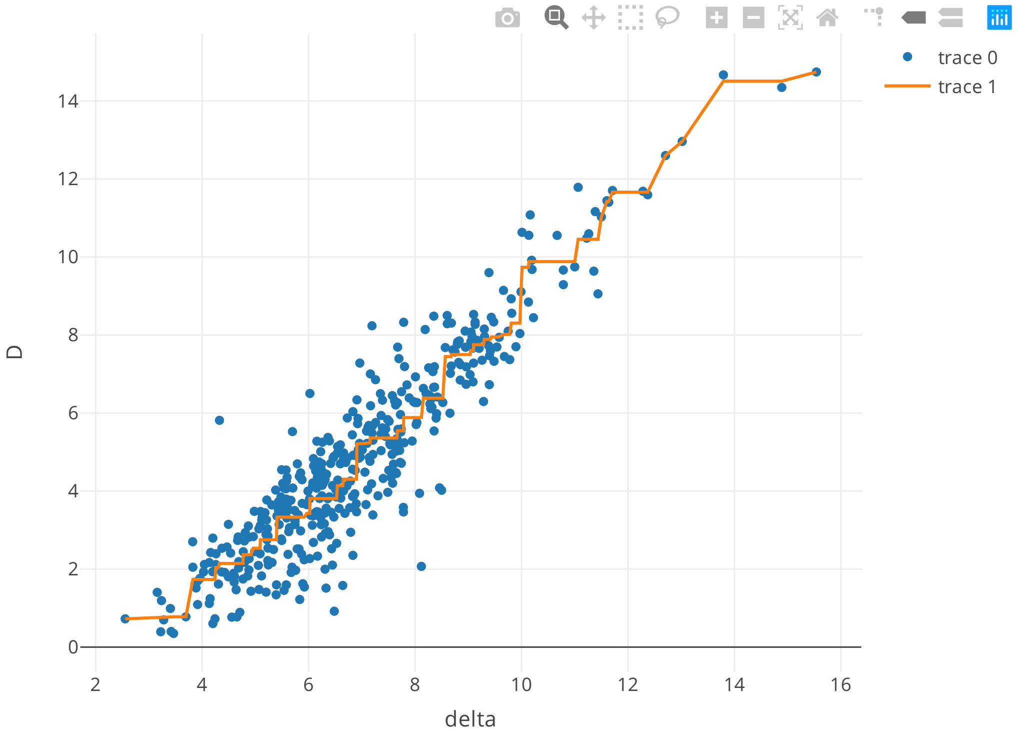


#3

```
library(readxl)
library(ggplot2)
library(plotly)
library(MASS)


df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)
scaled_df<-scale(df[,-c(1,2)])
d<-dist(scaled_df)
mds<-isoMDS(d,k=2)
```

```
## initial  value 19.856833 
## iter   5 value 16.319153
## iter  10 value 16.046215
## final  value 15.935476 
## converged
```

```
dt<-(mds$points)
dt_new<-as.data.frame(dt)
dt_new$League<-df$League
rownames(dt_new)<-df$Team


sh <- Shepard(d,dt)
delta <-as.numeric(d)
D<- as.numeric(dist(dt))

n=nrow(dt)
index=matrix(1:n, nrow=n, ncol=n)
index1=as.numeric(index[lower.tri(index)])

n=nrow(dt)
index=matrix(1:n, nrow=n, ncol=n, byrow = T)
index2=as.numeric(index[lower.tri(index)])



plot_ly()%>%
  add_markers(x=~delta, y=~D, hoverinfo ='text', text = ~paste('Obj1: ', df[index1,1],
                       '<br> Obj 2:', df[index2,1]))%>%add_lines(x=~sh$x, y=~sh$yf)
```

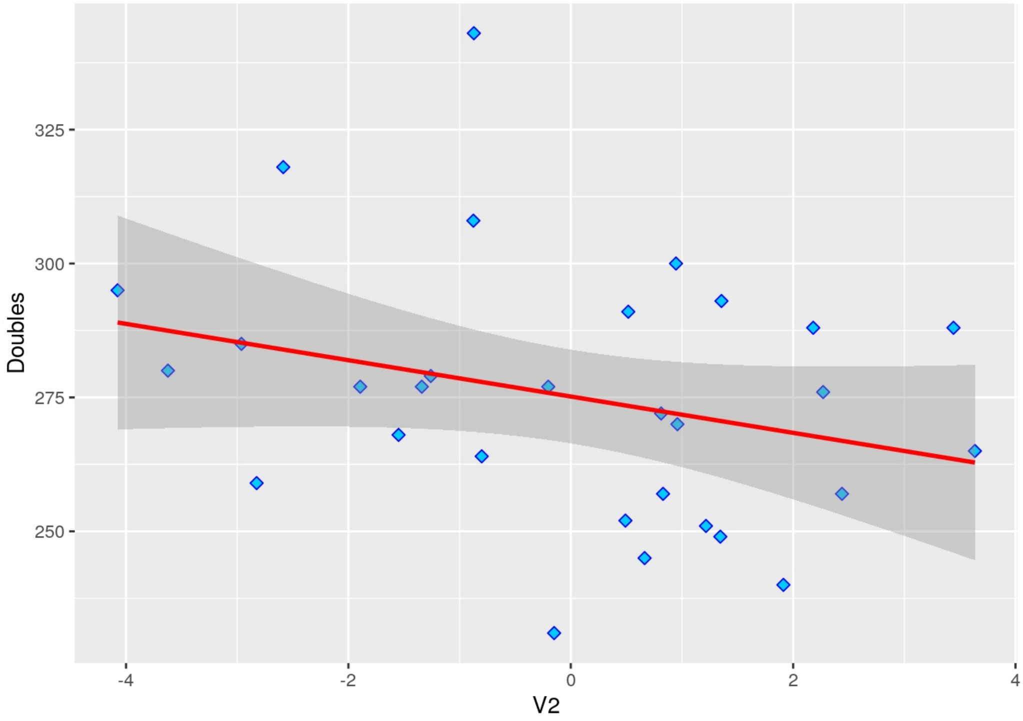


The data points away from the line are the ones that are problamatic.They are like Minnesota Twins and Arizona Diamondbacks, Oakland Athletics and Milwaukee Brewers) were hardest to plot from Shepard plot.

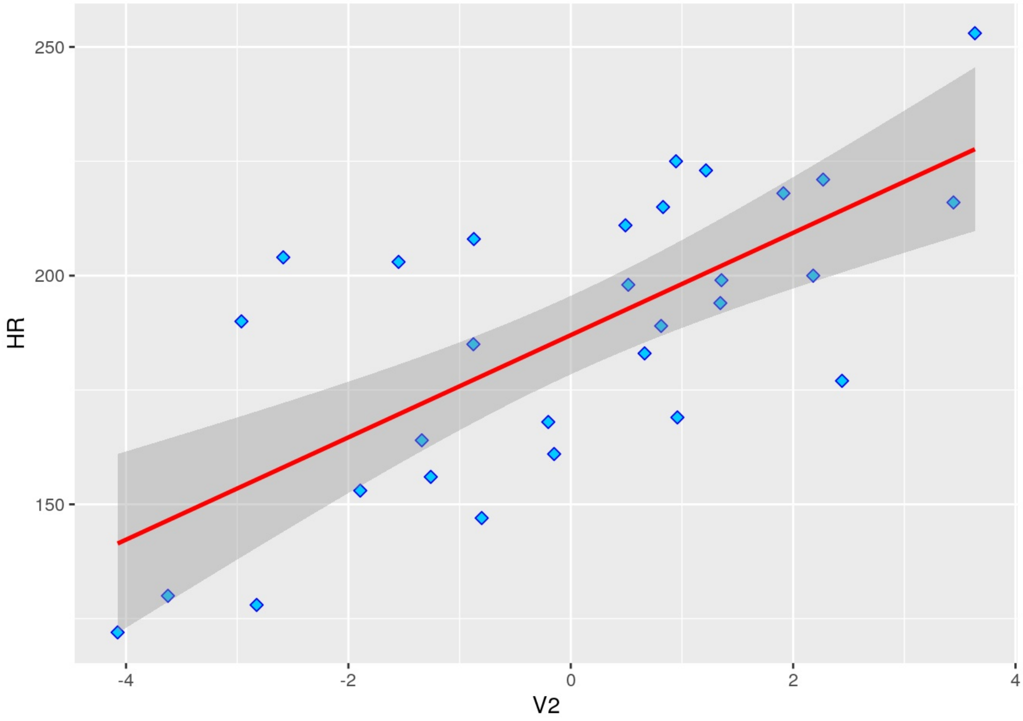
```
library(gridExtra)
num_df<-df[,-c(1,2)]
num_df$V2-dt_new$V2
```

```
## numeric(0)
```

```
num_df<-as.data.frame(num_df)
names(num_df)[8]<-"Doubles"
names(num_df)[9]<-"Triples"


ggplot(num_df,aes(x=dt_new$V2,y=Doubles))+geom_point(size=2,color="blue",pch=23,fill="#00CCFF")+stat_smooth(method
= "lm", col = "red")+xlab("V2")
```



```
ggplot(num_df,aes(x=dt_new$V2,y=HR))+geom_point(size=2,color="blue",pch=23,fill="#00CCFF")+stat_smooth(method = "l
m", col = "red")+xlab("V2")
```



Doubles and Home RUn could be observed as the two components that showed the strongest connection.This is true when considering the game itself especially for scoring. ###Appendix

```
## ----setup, include=FALSE,message=FALSE,warning=FALSE--------------------
knitr::opts_chunk$set(echo = TRUE)


## ----message=FALSE,warning=FALSE----------------------------------------
library(ggplot2)
library(plotly)
library(MASS)



olive<-read.csv("olive.csv",header=TRUE, sep=",")



p1<-ggplot(olive,aes(palmitic,oleic,col=linolenic))+geom_point()+labs(x="palmitic",y="oleic")+ggtitle("Plot of Pal
mitic on Oleic coloured by Linolenic")
```

```
p1

interval_cut <- cut_interval(olive$linolenic, 4)
p2<-ggplot(olive,aes(palmitic,oleic,col=interval_cut))+geom_point()+theme_light()+labs(x="palmitic",y="oleic")+ggt
itle("Plot of Palmitic on Oleic coloured by Linolenic Cut intervals")+theme(axis.text = element_text(colour = "blu
e"))
p2



## ---------------------------------------------------------------------------
p3<-s2<-ggplot(olive,aes(palmitic,oleic,col=interval_cut))+geom_point()+theme_light()+labs(x="palmitic",y="oleic")
+geom_point(size=3)+ggtitle("Plot of Palmitic on Oleic coloured by Linolenic")+theme(axis.text = element_text(colo
ur = "blue"))
p3



## ---------------------------------------------------------------------------
p4<-s2<-ggplot(olive,aes(palmitic,oleic))+geom_point(aes(size=interval_cut))+theme_light()+labs(x="palmitic",y="ol
eic")+geom_point()+ggtitle("Plot of Palmitic on Oleic coloured by Linolenic")+theme(axis.text = element_text(colou
r = "blue"))
p4


## ---------------------------------------------------------------------------
angleplot<-ggplot(olive,aes(x=palmitic,y=oleic)) +
  geom_point() +geom_spoke(aes(angle = as.numeric(cut_interval(linolenic, 4))*pi/2), radius = 50)
angleplot


## ---------------------------------------------------------------------------
p6<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=as.numeric(Region)))+labs(x="oleic",y="eicosenoic")+geom_point(siz
e=2)+ggtitle("oleic on eicosenoic coloured by Region")
p6


## ---------------------------------------------------------------------------
p7<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=as.factor(Region)))+labs(x="oleic",y="eicosenoic")+geom_point(size
=2)+ggtitle("oleic on eicosenoic coloured by Region")
p7


## ---------------------------------------------------------------------------
discretelionelic<-cut_interval(olive$linolenic, 3)
discretepalmitic<-cut_interval(olive$palmitic, 3)
discretepalmitoleic<-cut_interval(olive$palmitoleic, 3)

p8<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=discretelionelic,shape=discretepalmitic,size=discretepalmitoleic))
+geom_point()+ggtitle("oleic vs eicosenoic with linolenic, palmitic,palmitoleic")
p8


## ---------------------------------------------------------------------------
p9<-ggplot(olive,aes(x=oleic,y=eicosenoic,color=Region,shape=discretepalmitic,size=discretepalmitoleic))+geom_poin
t()+ggtitle("oleic vs eicosenoic with linolenic, palmitic,palmitoleic")
p9


## ----message=FALSE,warning=FALSE----------------------------------------------
library(plotrix)
library(plotly)

mytable <- table(olive$Area)
dframe<-as.data.frame(mytable)

#pie3D(dframe$Freq,labels=dframe$Var1,explode=0.1,main="Pie Chart of Area Proportion ")

p10 <- plot_ly(dframe, labels = ~Var1, values = ~Freq, type = 'pie',textinfo = "none", hoverinfo = "text",text=~Va
r1) %>%
  layout(title = 'Pie chart of Area Proportion',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

p10


## ---------------------------------------------------------------------------
```

```
p11<-ggplot(olive, aes(x=linoleic, y=eicosenoic,col="red") ) +geom_density_2d(position="identity")+theme_dark()
p11


## ----message=FALSE,warning=FALSE-------------------------------------------
library(readxl)
library(ggplot2)
library(plotly)
library(MASS)


df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)


## --------------------------------------------------------------------------
library(readxl)

df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)
df<-data.frame(df)

scaled_df<-scale(df[,-c(1,2)])
d<-dist(scaled_df)
mds<-isoMDS(d,k=2)

dt<-(mds$points)
dt_new<-as.data.frame(dt)
dt_new$League<-df[,2]
dt_new$Team<-df[,1]



plot_ly(dt_new, x=~V1, y=~V2, type="scatter", hovertext=~Team, color=~League, colors=c("green", "red"))


## ----message=FALSE,warning=FALSE-------------------------------------------
library(readxl)
library(ggplot2)
library(plotly)
library(MASS)


df<-read_xlsx("baseball-2016.xlsx",col_names=TRUE)
scaled_df<-scale(df[,-c(1,2)])
d<-dist(scaled_df)
mds<-isoMDS(d,k=2)

dt<-(mds$points)
dt_new<-as.data.frame(dt)
dt_new$League<-df$League
rownames(dt_new)<-df$Team


sh <- Shepard(d,dt)
delta <-as.numeric(d)
D<- as.numeric(dist(dt))

n=nrow(dt)
index=matrix(1:n, nrow=n, ncol=n)
index1=as.numeric(index[lower.tri(index)])

n=nrow(dt)
index=matrix(1:n, nrow=n, ncol=n, byrow = T)
index2=as.numeric(index[lower.tri(index)])



plot_ly()%>%
  add_markers(x=~delta, y=~D, hoverinfo ='text', text = ~paste('Obj1: ', df[index1,1],
                      '<br> Obj 2:', df[index2,1]))%>%add_lines(x=~sh$x, y=~sh$yf)


## ----message=FALSE,warning=FALSE-------------------------------------------
library(gridExtra)
num_df<-df[,-c(1,2)]
num_df$V2-dt_new$V2
num_df<-as.data.frame(num_df)
names(num_df)[8]<-"Doubles"
names(num_df)[9]<-"Triples"
```

```
ggplot(num_df,aes(x=dt_new$V2,y=Doubles))+geom_point(size=2,color="blue",pch=23,fill="#00CCFF")+stat_smooth(method = "lm", col = "red")+xlab("V2")



## ---------------------------------------------------------------------------

ggplot(num_df,aes(x=dt_new$V2,y=HR))+geom_point(size=2,color="blue",pch=23,fill="#00CCFF")+stat_smooth(method = "lm", col = "red")+xlab("V2")
```