Lab 2

Group 22 - balra340, tejma768

20 September 2018

Assignment 1

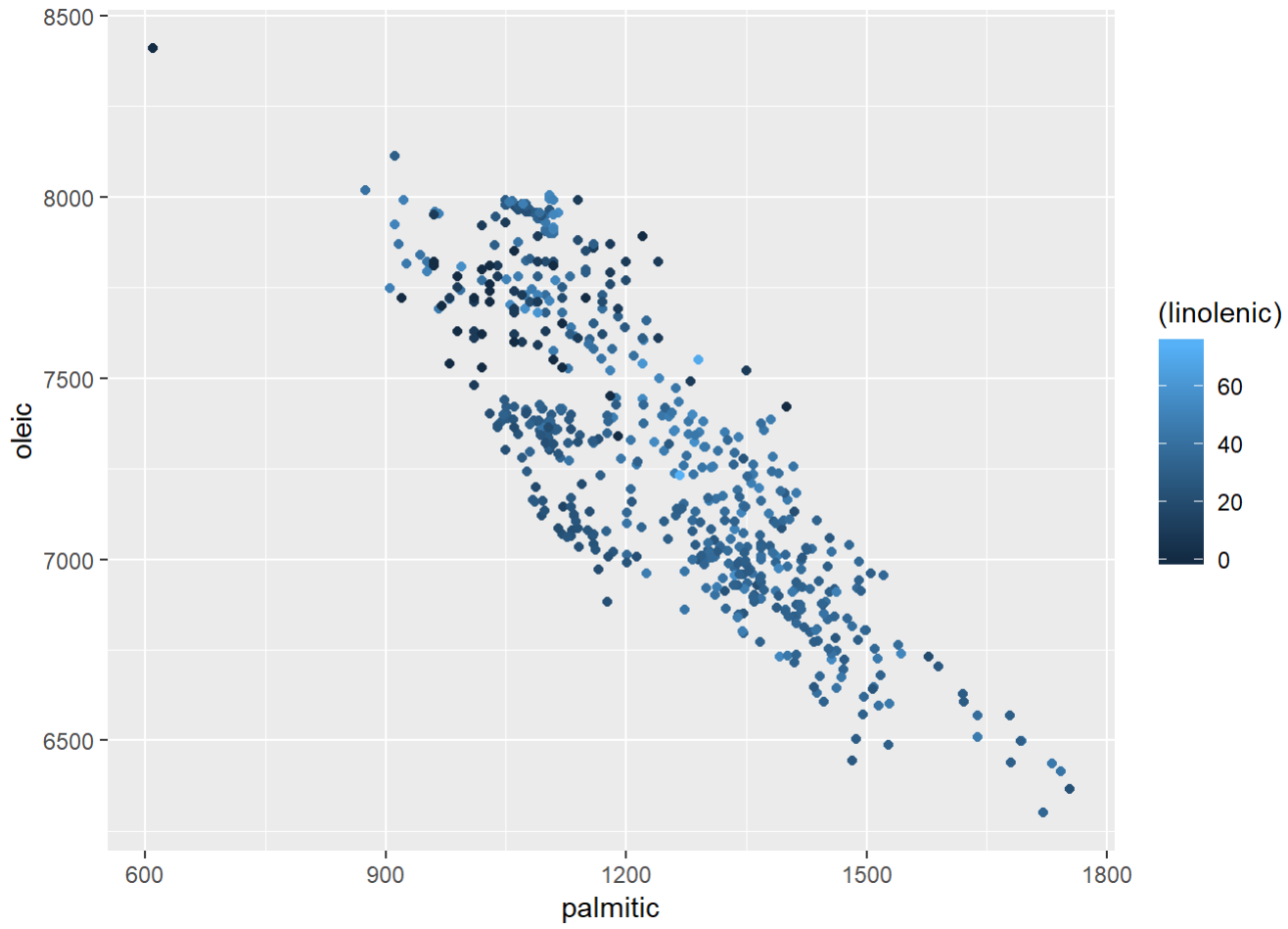
For this part of assignment we use a the data set **olive.csv** which contains information about the contents of olive oil from different regions of Italy.

```
library(ggplot2)

olive_data <- read.csv2("olive.csv",sep = ",")
```

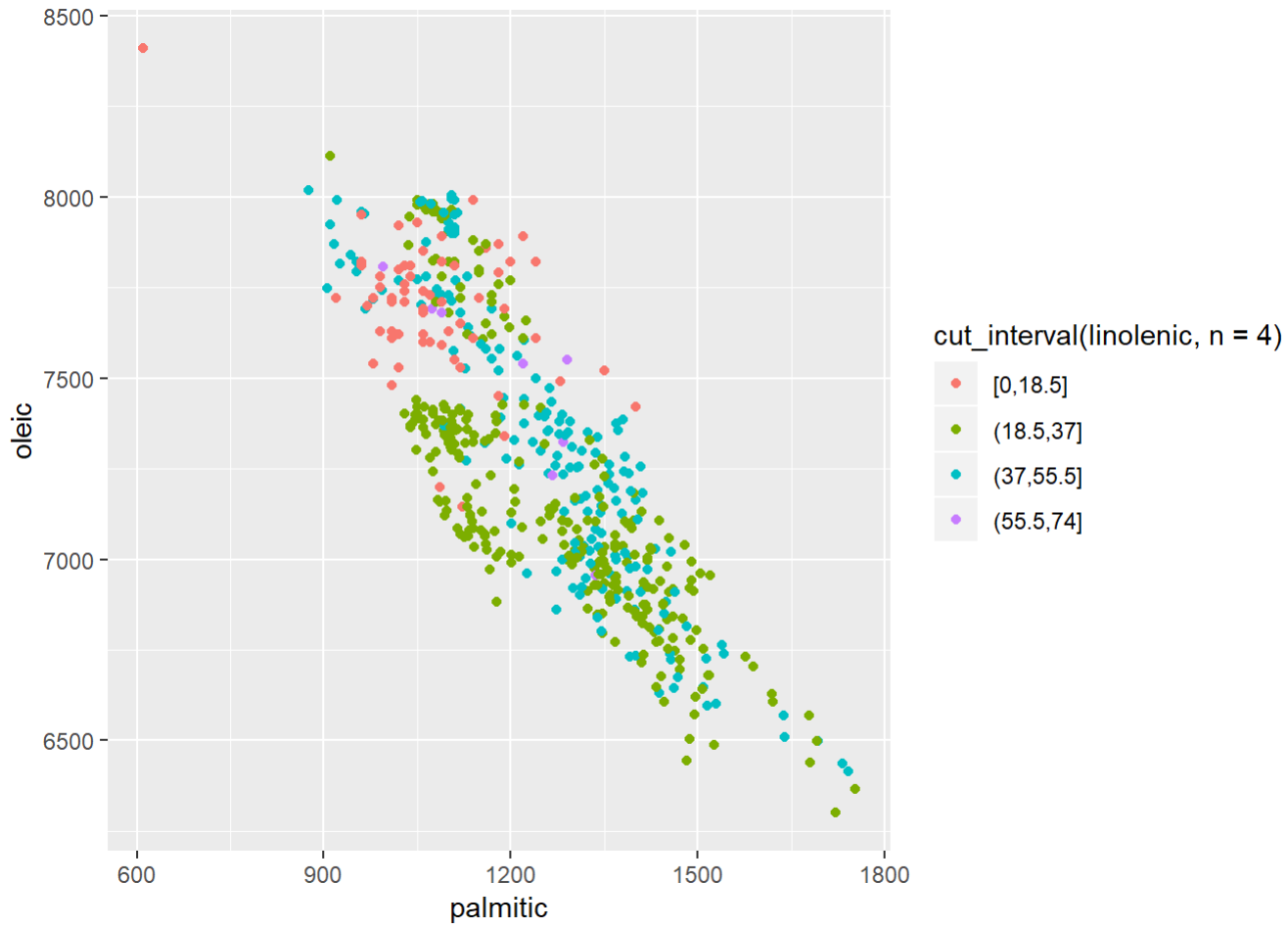
1.1

In this question, we create a scatter plot to show a dependence of **palmitic on oleic** and the observations are coloured by **linolenic**. And also a similar plot in which the colour parameter is divided into four classes.

```
#normal one

ggplot(data = olive_data) + geom_point(aes(x = palmitic, y = oleic , colour = (linolenic)))
```



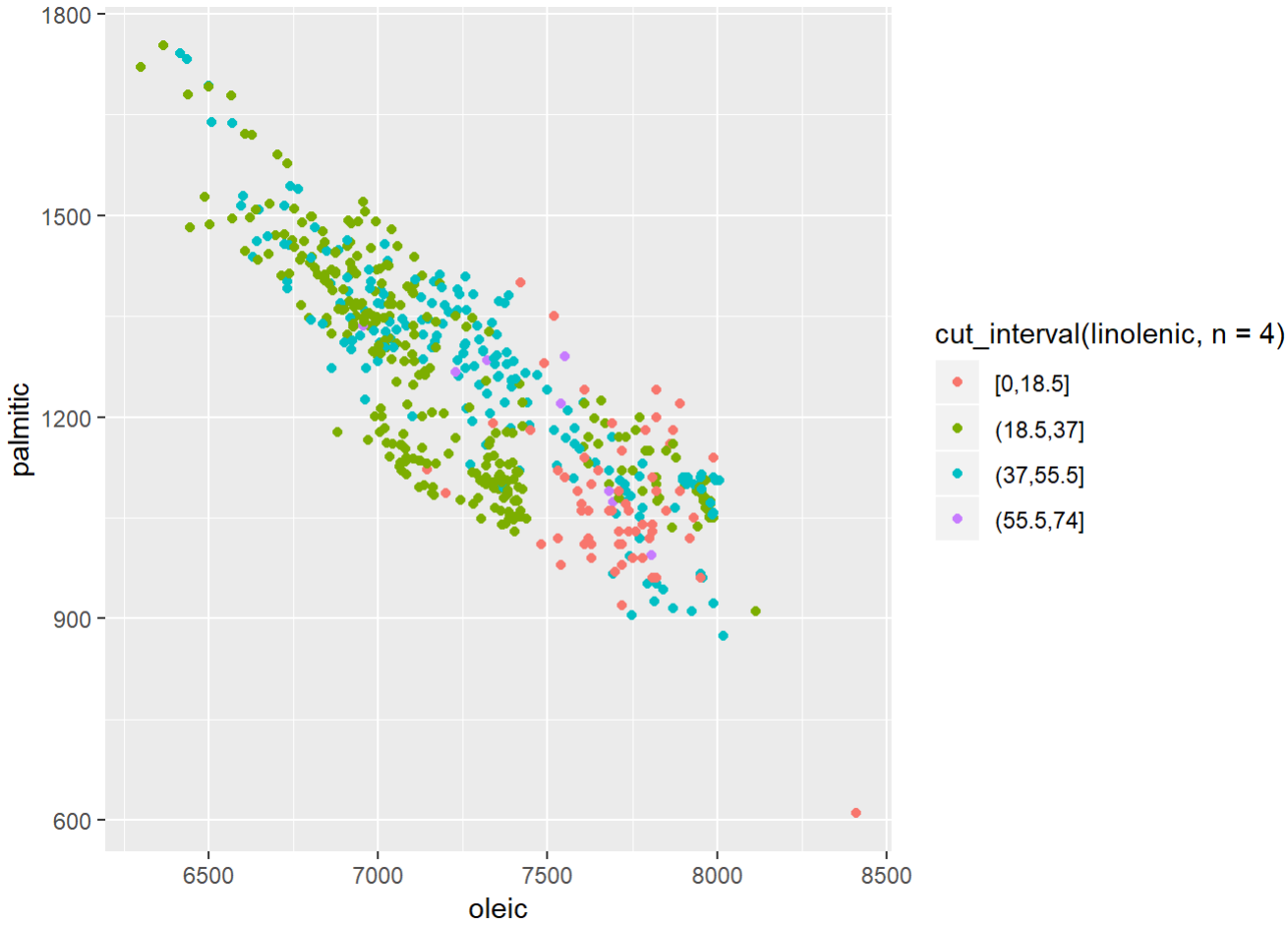
```
#using cut_interval

ggplot(data = olive_data) + geom_point(aes(x = palmitic, y = oleic , colour = cut_interval(linol
enic,n=4 )))
```



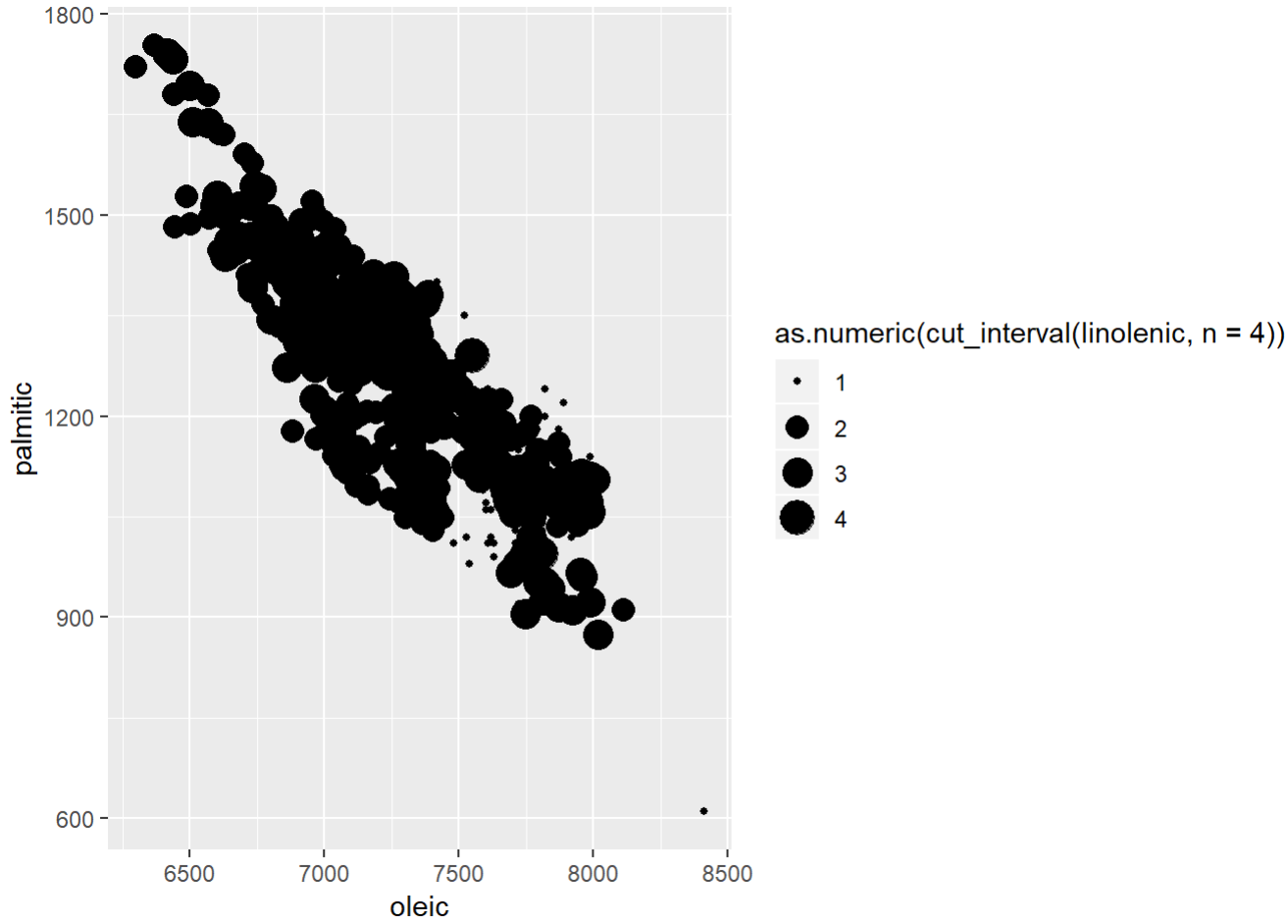
The plot where the colour is classifed into four groups is much easier to analyze when compared to the normal ones. The classification helps in distinguishing the points easily. The benefit of pre attentive perception mechanism is exhibited in this. The channel capacity for hue is 10 levels but in our case we are using only 4. So it is easier to visualize.

1.2 Now we create scatterplots of palmitic vs oleic in which linolenic is classified into four and assigned to color , size and orientation angle.

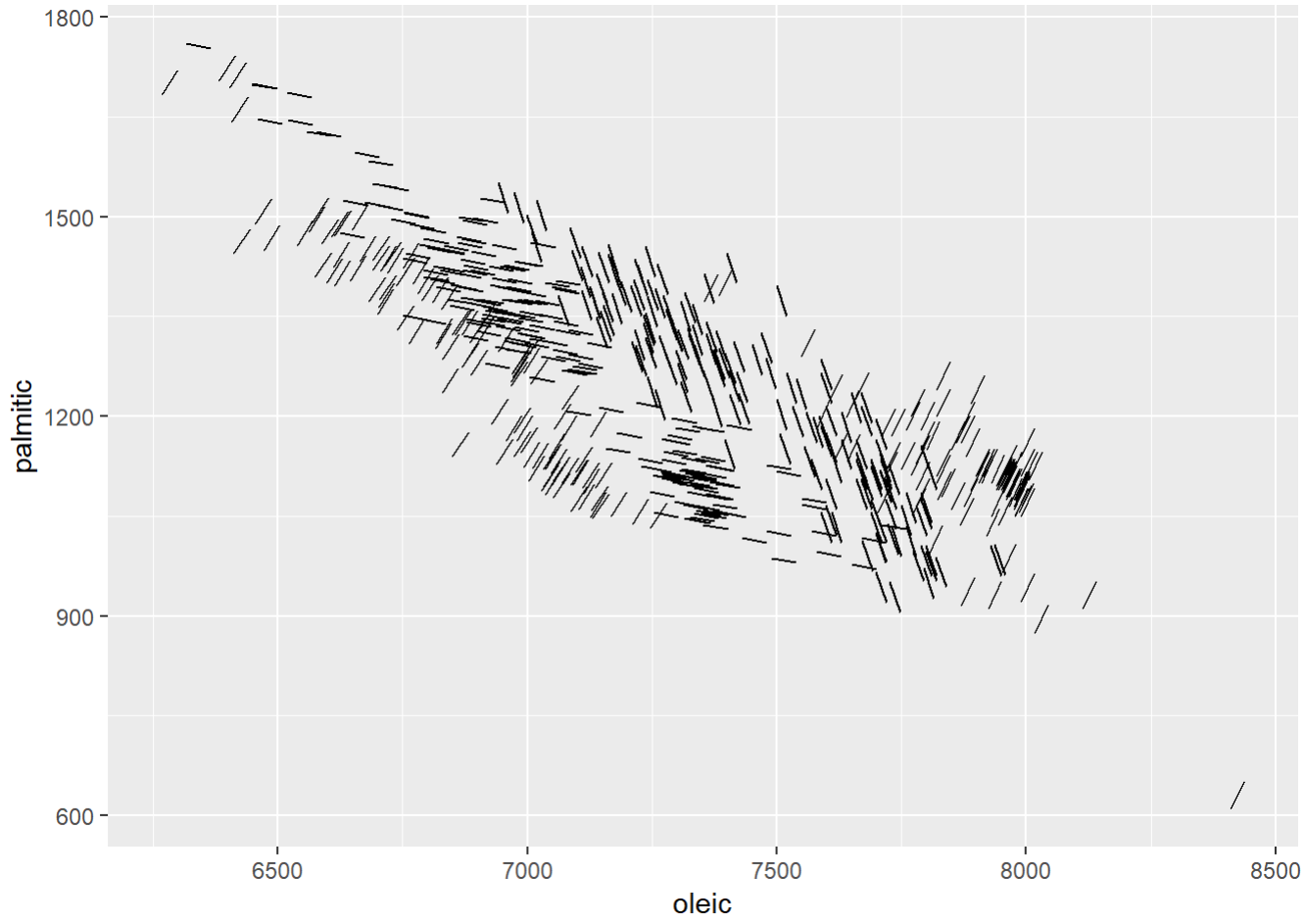
```
#a

ggplot(data = olive_data) + geom_point(aes(x = oleic, y = palmitic , colour = cut_interval(linol
enic,n=4 )))
```




```
#b

ggplot(data = olive_data) + geom_point(aes(x = oleic, y = palmitic , size = as.numeric(cut_inter
val(linolenic,n=4))))
```




```
#This is not good. Too difficult to distinguish


#c

a <- as.numeric(cut_interval(olive_data$linoleic,4))

ggplot(data = olive_data,aes(x = oleic, y = palmitic )) + geom_spoke(aes(angle = a ) , radius =
50 )
```



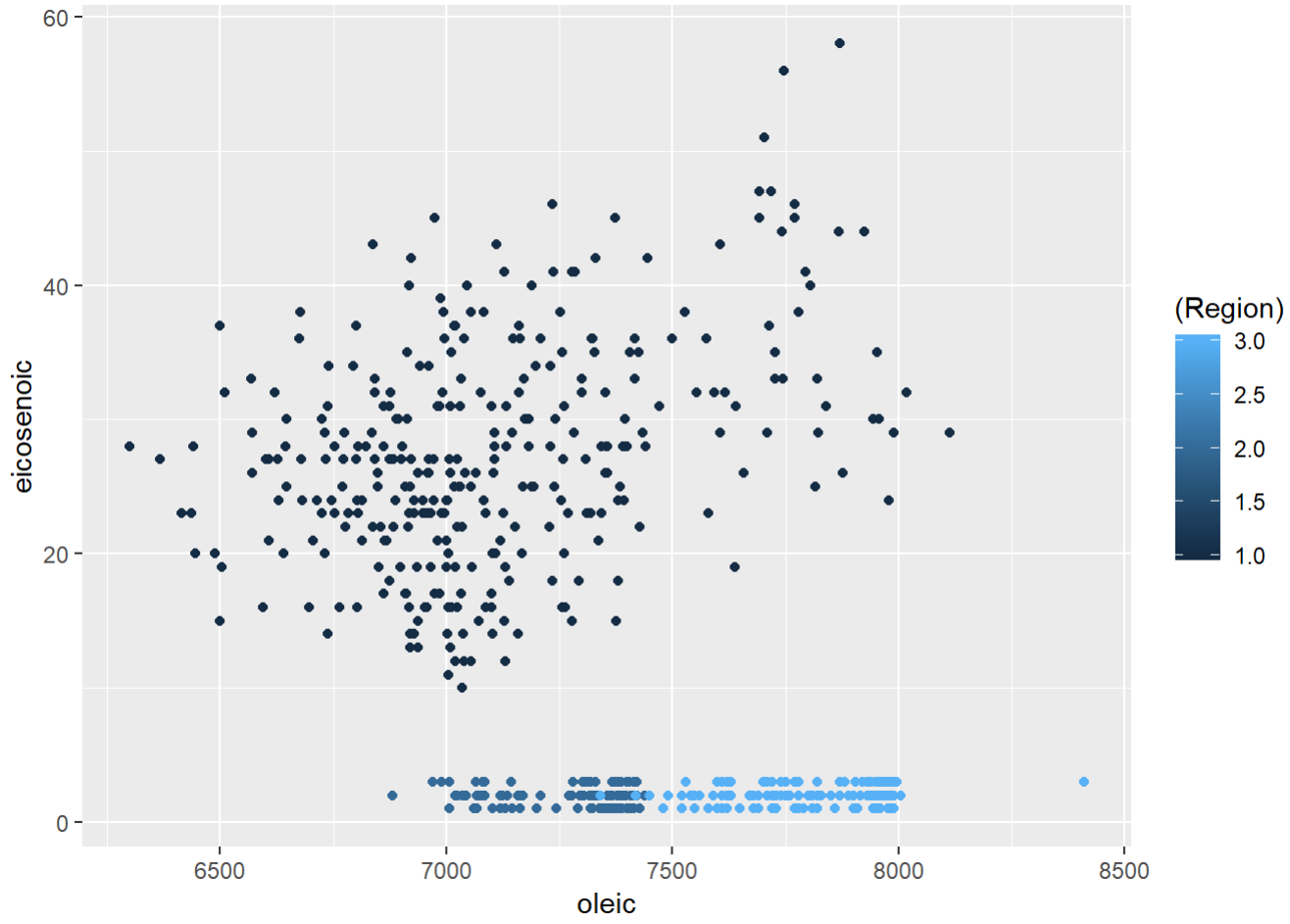

```
#this is still bad
```

It was difficult in the second and third case where the classification was assigned to size and orientation angle. When we use size , we can observe only 4-5 levels [2.2 bits] which is lesser than line orientation [3 bits].In the case of hue its 10 levels [3.1 bits] and is considered as the best out of the three. Hence the second case is the toughest out of all the three.

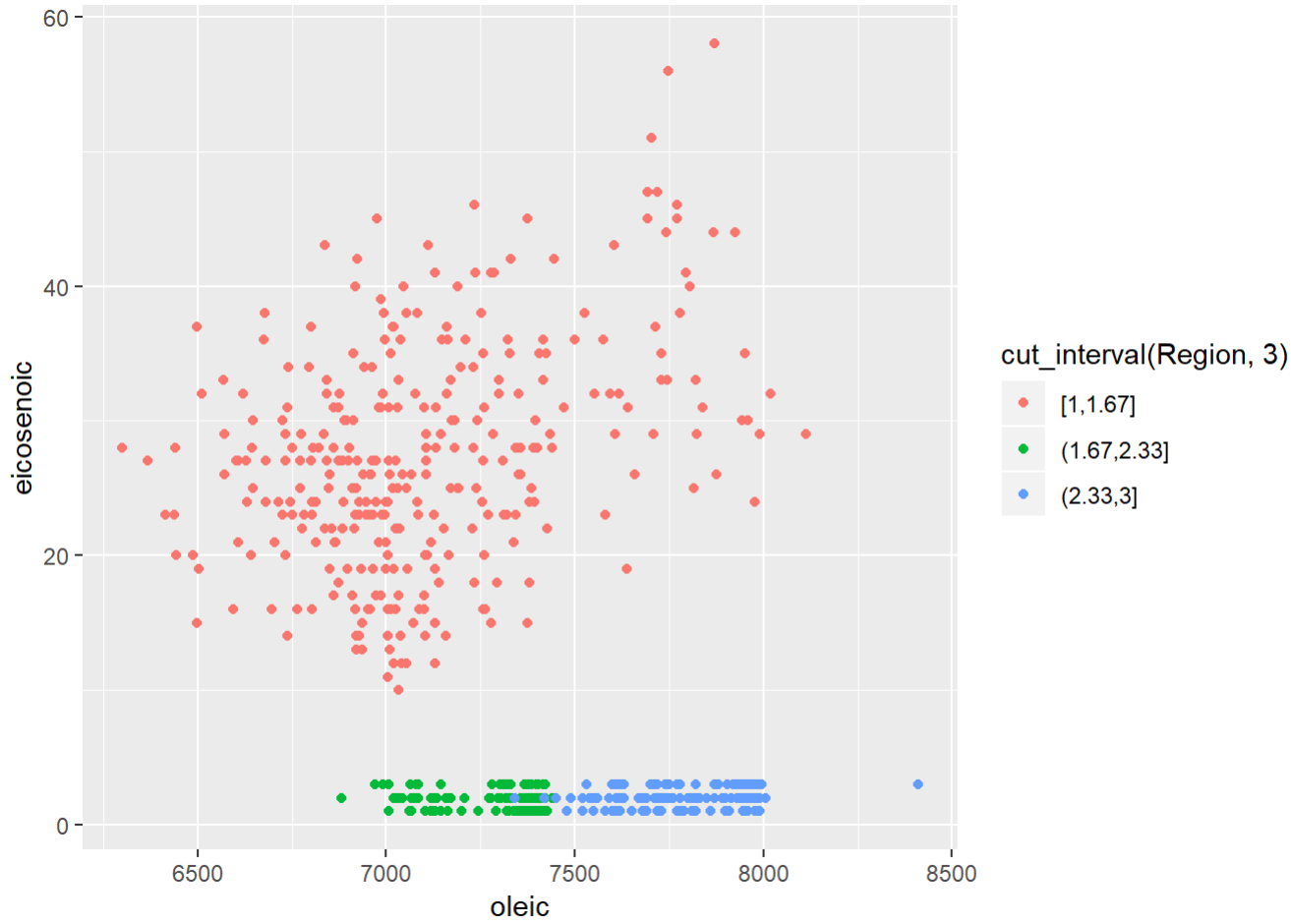
1.3

Now we create a scatterplot of oleice vs eicosenoic in which color is defined by region and a similar plot with color as cateogorical variable.

```
ggplot(data = olive_data) + geom_point(aes(x = oleic, y = eicosenoic , colour = (Region)))
```




```
ggplot(data = olive_data) + geom_point(aes(x = oleic, y = eicosenoic , colour = cut_interval(Reg
ion,3)))
```




```
# Nw its much easier as we use different colours . Yes it made its possible
```

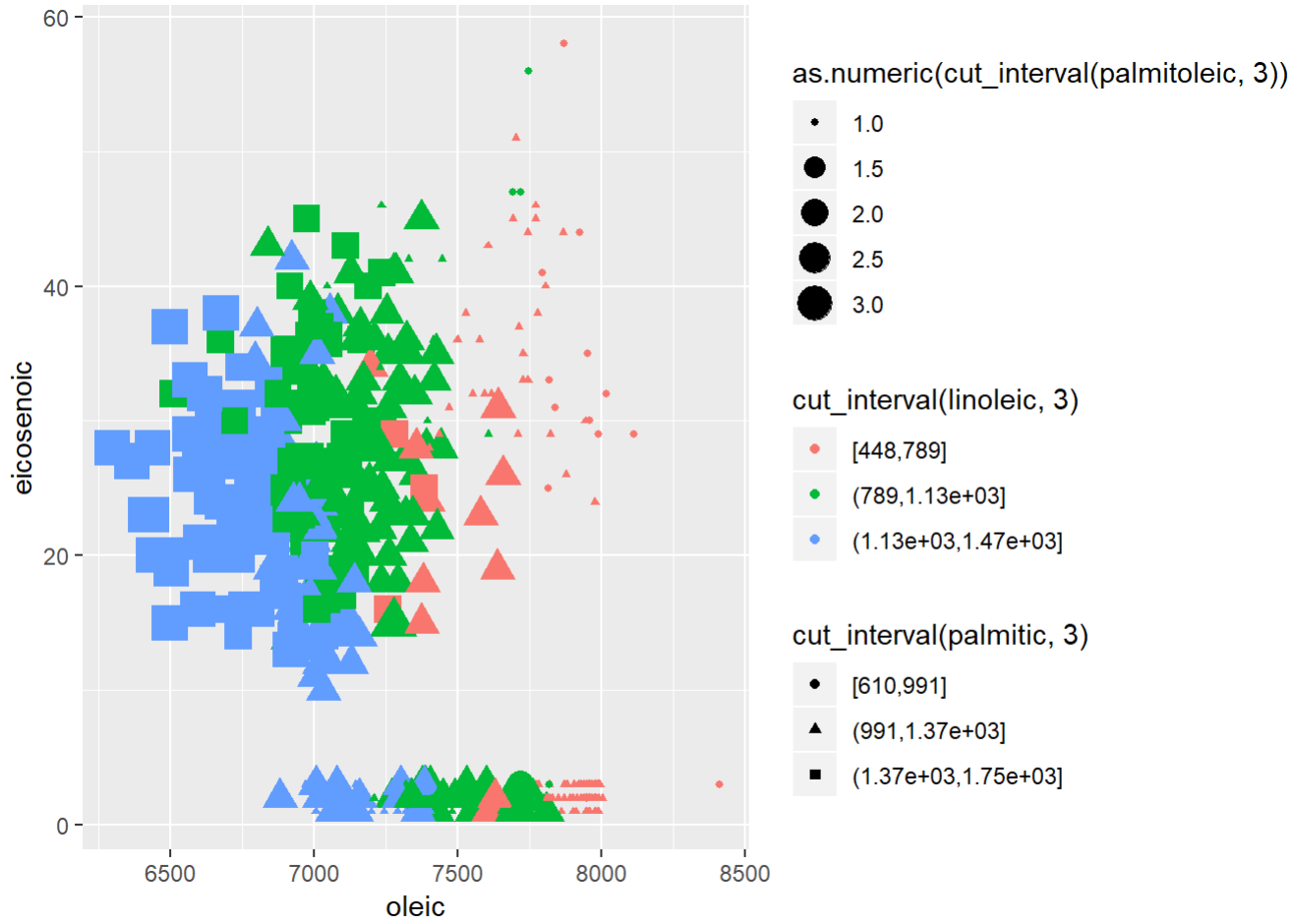
The first plot represents different regions with difference in brightness. This cannont work if the number of regions is more than 5 as human eye can only identify 5 levels of brightness.

Whereas in the second case cateograzing makes is easy to distinguis as every catoegary has a different colour and human eye can understand upto 10 colours. This is achieved by pre attentive mechanism.

1.4

In this we create a scatterplot of oleic vs eicosenoic., colour is classified into three classes of linolenic , shape is defined by 3 classes of palmitic and size is grouped by 3 classes of palmitoleic.

```
ggplot(data = olive_data) + geom_point(aes(x = oleic, y = eicosenoic , colour = cut_interval(lin
oleic,3) ,size = as.numeric(cut_interval(palmitoleic,3)) ,  shape = cut_interval(palmitic,3)  ))
```

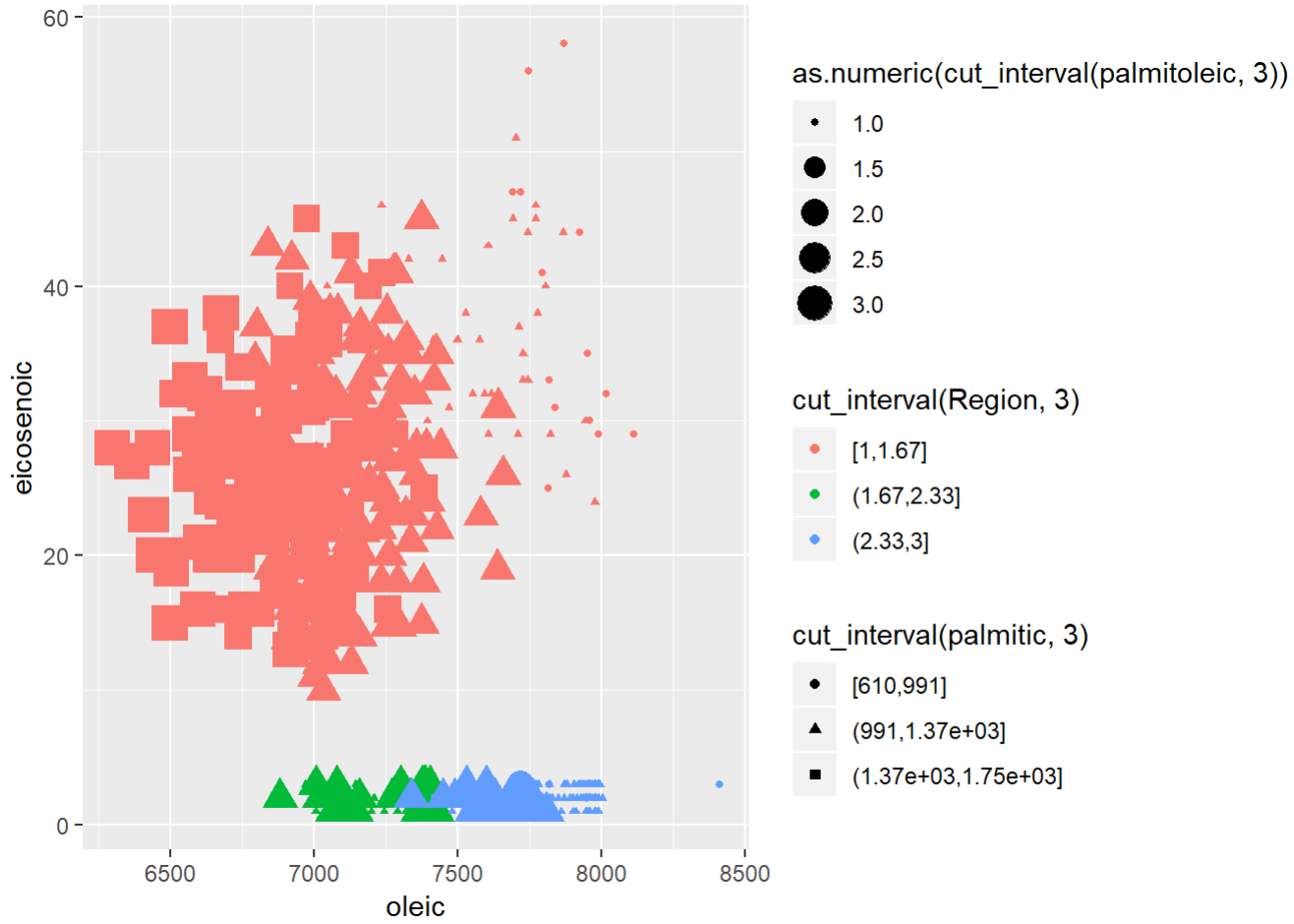


There is too much detailing in the graph which makes the analysis difficult. This leads to occlusion which means overlapping of objects on another objects in the graph. And also we use combining too many metrics, total of 27 because of which pre attentive mechanism fails. The channel capacities sum up to a values greater than the value which is ideal.

1.5

This is exactly like the previous question except for the colour parameter which is grouped by Region.

```
ggplot(data = olive_data) + geom_point(aes(x = oleic, y = eicosenoic , colour = cut_interval(Reg
ion,3) ,size = as.numeric(cut_interval(palmitoleic,3)) ,  shape = cut_interval(palmitic,3)  ))
```

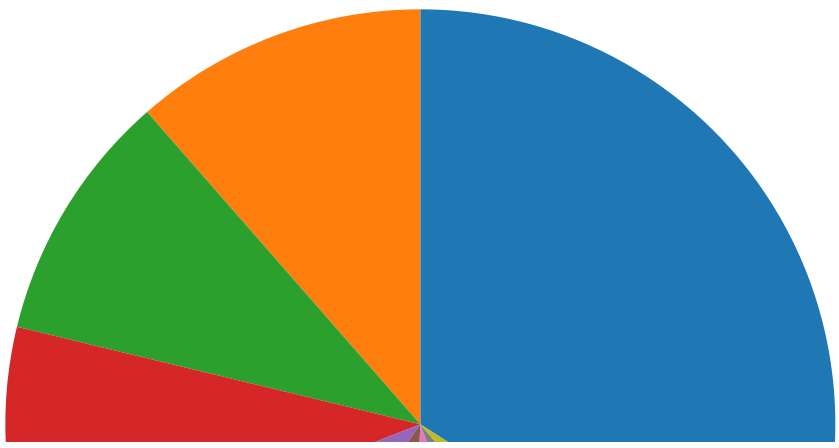


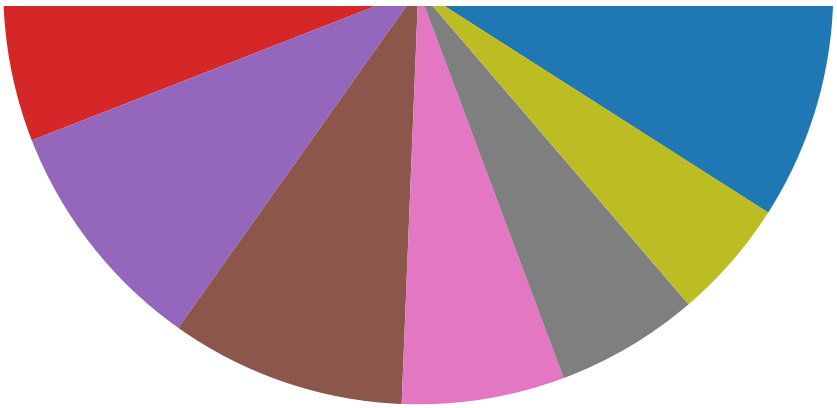
In this case we are able to find the boundaries easily because according to Triesman's theory in this plot, the color and shape are processed parallely becaus of whivh they are considered as an individual feature map.

1.6

In this question we create a pie chart using plotly to show proportions of oil coming from different areas.

```
library(plotly)

p <- plot_ly(olive_data, labels = ~Area, values = ~oleic, type = 'pie',textinfo = "none")%>%layo
ut(showlegend = F)

p
```



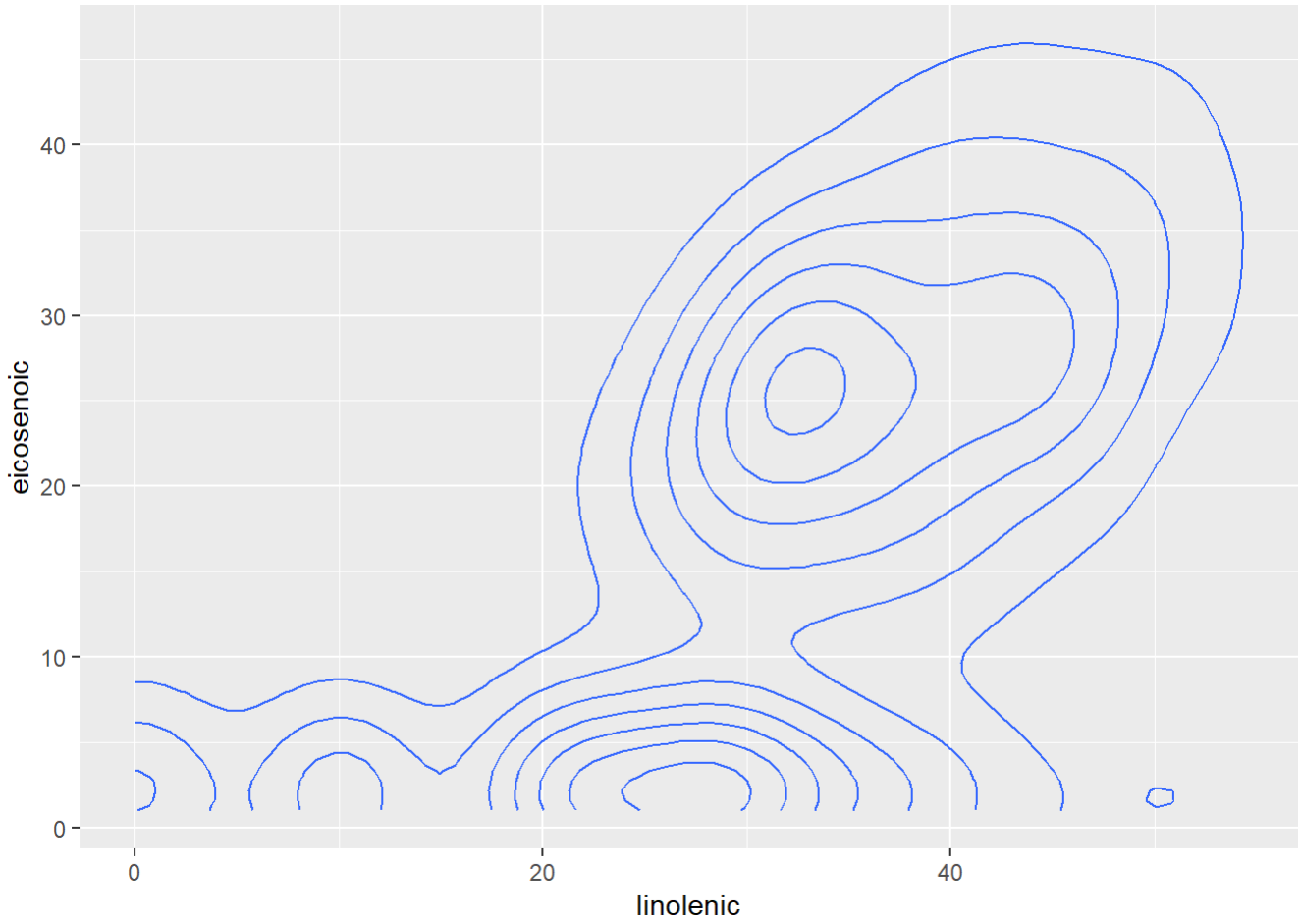


Without the labels its difficult to understand the aesthetic mapping.Everytime we have to hover over a particular sector to understand which belongs to which area.

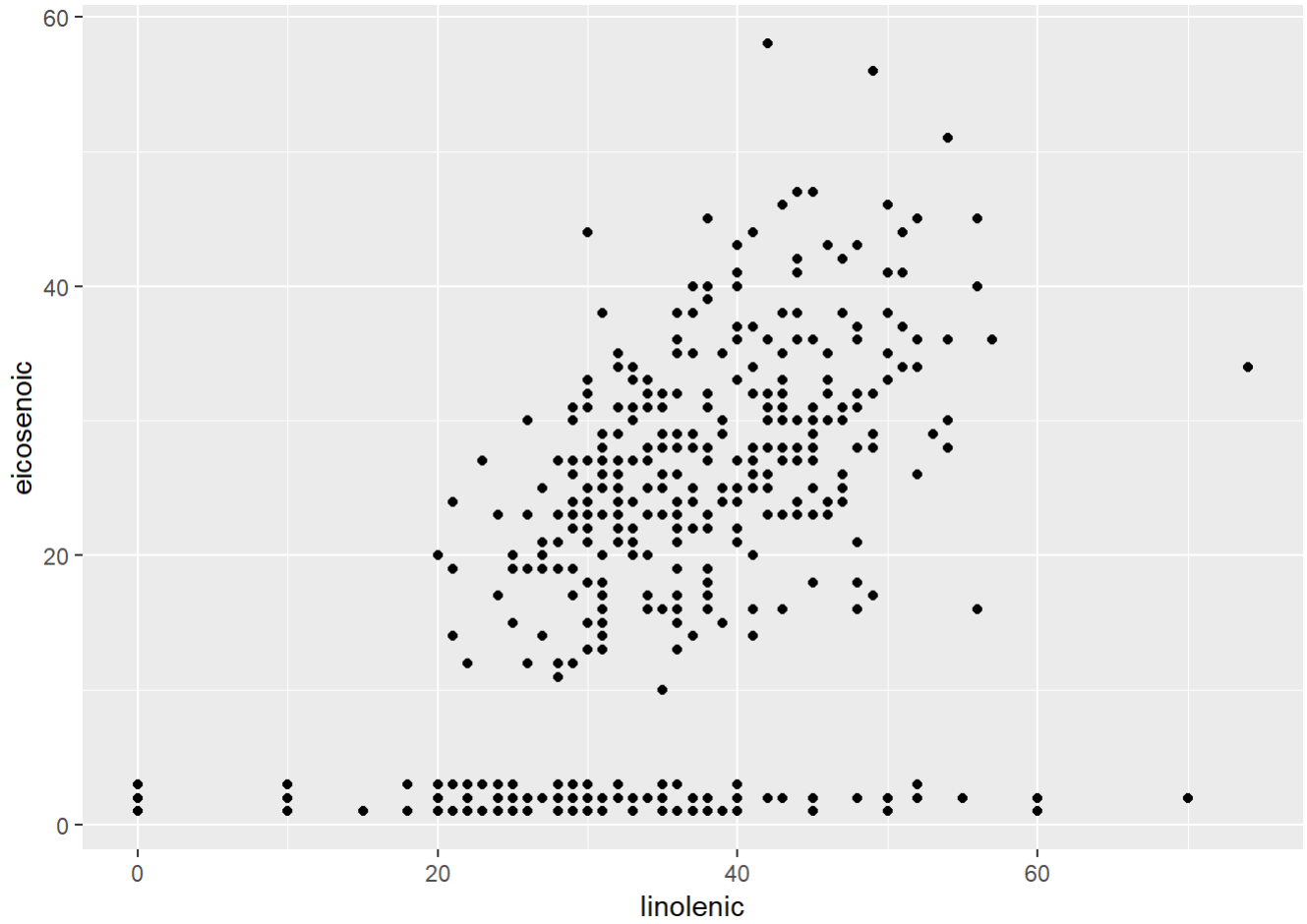
1.7

In this we create a 2d density plot and scatter plot to show the dependance of linoleic vs eicosenoic

```
ggplot(data = olive_data) + geom_density_2d(aes(x = linolenic , y = eicosenoic))
```




```
ggplot(data = olive_data) + geom_point(aes(x = linolenic, y = eicosenoic ))
```



The density plots smoothens the noises which leads to loss of few data points. Acoording to the density plot the y axis values are just shown in 40. But from the scatter plot we can see there are values close to 60 also.

Assignment 2

For this question we use the data set **baseball-2016**.

2.1

```
library("readxl")

baseball <- read_excel("baseball-2016.xlsx")

baseball_scaled <- as.data.frame( scale(baseball[,3:28])) #scaled format

baseball_scaled <- cbind(baseball[,1:2],baseball_scaled) #addig the first 2 columns
```

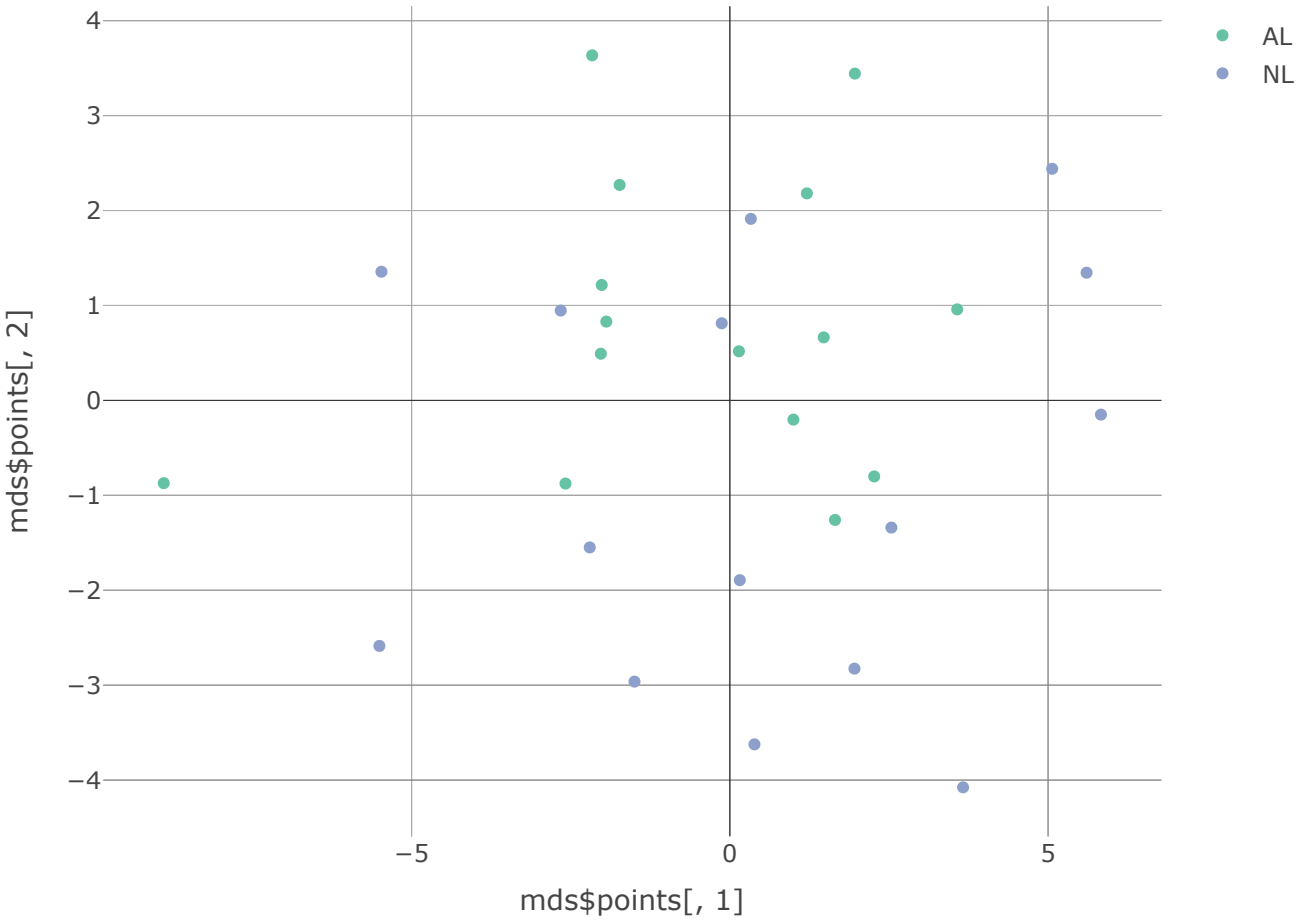
Yes it is necessary to perform scaling so the distance can be minimised which can be helpful in performing MDS.

2.2

```
library(plotly)
library("MASS")

distance <- dist(baseball_scaled[,3:28] , method = "minkowski")

mds <- isoMDS(distance,p=2)
```

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

```
plot_ly(type = "scatter",data = as.data.frame(mds),x=~mds$points[,1], y=~mds$points[,2] ,
        mode = "markers",color = ~baseball$League)
```

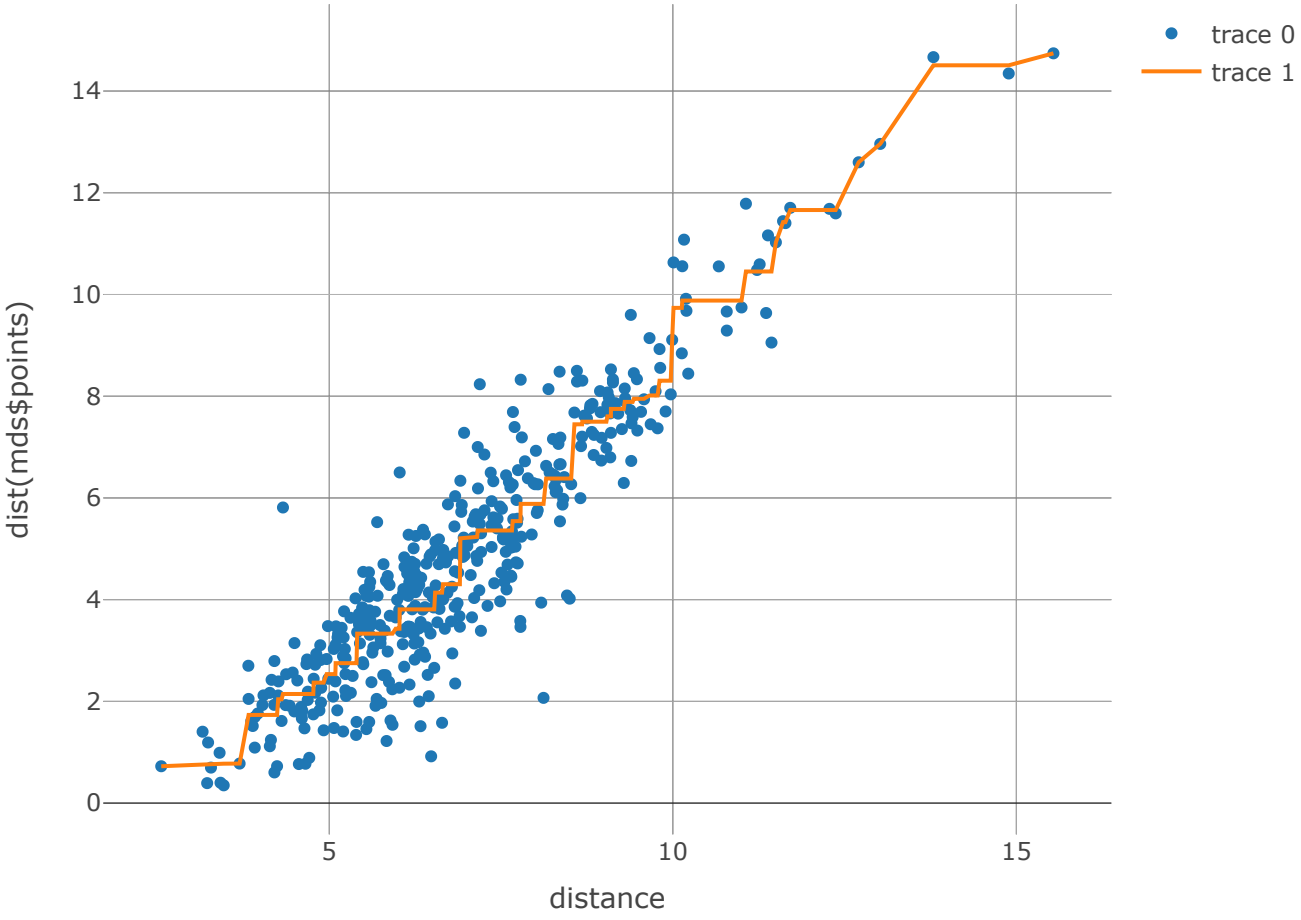


From the plot we can see the points representing the leagues are not actually closer which exhibits the difference in the leagues. AL points are mostly present in the center where as NL points are away from it.The y component helps in distinguishing the two leagues better.And the outliers for AL is Boston Red Fox since its far from the center and the outlier for NL is Los angeles dodgers as it is close to the center.

2.3

For this part of the queestion we are supposed to make a shepard plot.

```
library(dplyr)
shepard <- Shepard(distance, mds$points)


n  <- nrow(mds$points)
index  <-  matrix(1:n, nrow=n, ncol=n)
ind1  <-  as.numeric(index[lower.tri(index)])

n  <-  nrow(mds$points)
index  <-  matrix(1:n, nrow=n, ncol=n, byrow = TRUE)
ind2 <-  as.numeric(index[lower.tri(index)])

row.names(baseball) <- baseball %>% pull(Team)




plot_ly(type = "scatter",data = as.data.frame(mds),x=~distance, y=~dist(mds$points) ,
        mode = "markers",hoverinfo = 'text' , text = ~paste( rownames(baseball)[ind1],",",
                                                             rownames(baseball)[ind2])) %
>%
  add_trace(x = ~shepard$x , y = ~shepard$yf , mode = "line")
```



The MDS fit was decent as it almost follows the trend but there was some trouble in plotting Minnesota Twins vs Arizona Diamondbacks, Oakland Athletics VS Milwaukee Brewers

2.4 Produce series of scatterplots in which you plot the MDS variable that was the best in the differentiation between the leagues in step 2 against all other numerical variables of the data. Pick up two scatterplots that seem to show the strongest (positive or negative) connection between the variables

```
data1 <- cbind(baseball_scaled, mds$points[,2])
data1 <- as.data.frame(data1)

p1 <- ggplot(data=data1, aes(x=IBB, y=data1[,11])) + geom_point() +
    labs(y=colnames(data1)[11] )


p2 <- ggplot(data=data1, aes(x=IBB, y=data1[,12])) + geom_point() +
    labs(y=colnames(data1)[12] )
```

From the scatterplot, strongest positive connection with MDS variable Y was found to be Home Run variable, and the strongest connection with MDS variable Y was of riples variable. On searching in google, we got to know that both Home Run and Triples are important in baseball for scoring runs. Home runs are among the most important events, Tripples are becoming a little rare. In relation to the process of making runs, the chosen variable is highly influenced by these variables.