Visualization Lab2 Group2

Assignment 1

```
olive=read.csv("olive.csv")
theme_set(theme_bw())
```

1. Task 1

```
p1<- ggplot()  +
  geom_point(data=olive,aes(x=oleic,y=palmitic,color=linoleic)) +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="I",
       subtitle="The dependence of Palmitic on Oleic",
       caption="Data:NMMAPS",
       tag="Fig.1.1") +
  scale_colour_gradientn(colours = c("#66CCFF44", "#66CCFF"))
#Divide a continuous variable into three equal-sized groups
x<-data.frame(olive,discretized=cut_interval(olive$linoleic, 4))
p2<-ggplot()  +
  geom_point(data=x,aes(x=oleic,y=palmitic,color=discretized)) +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="II",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.1.2") #+
```

Fig.1.1

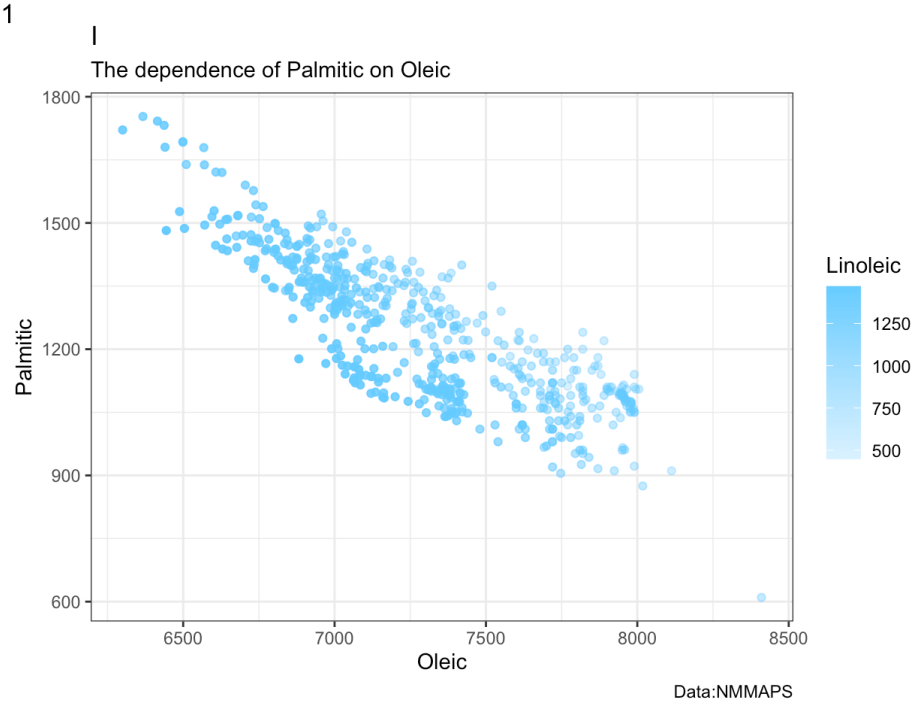
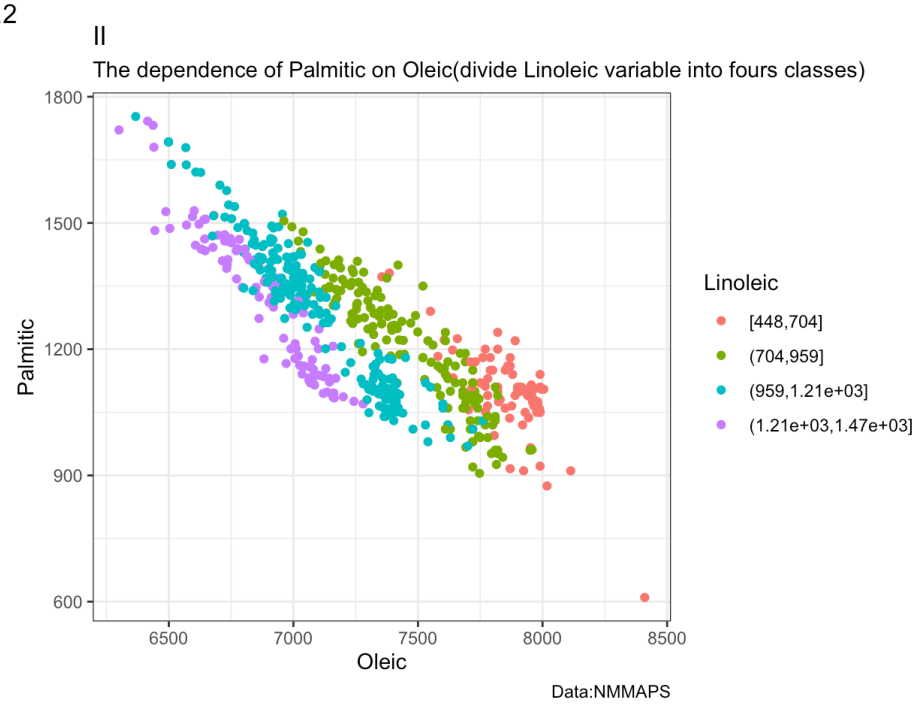


Fig.1.2

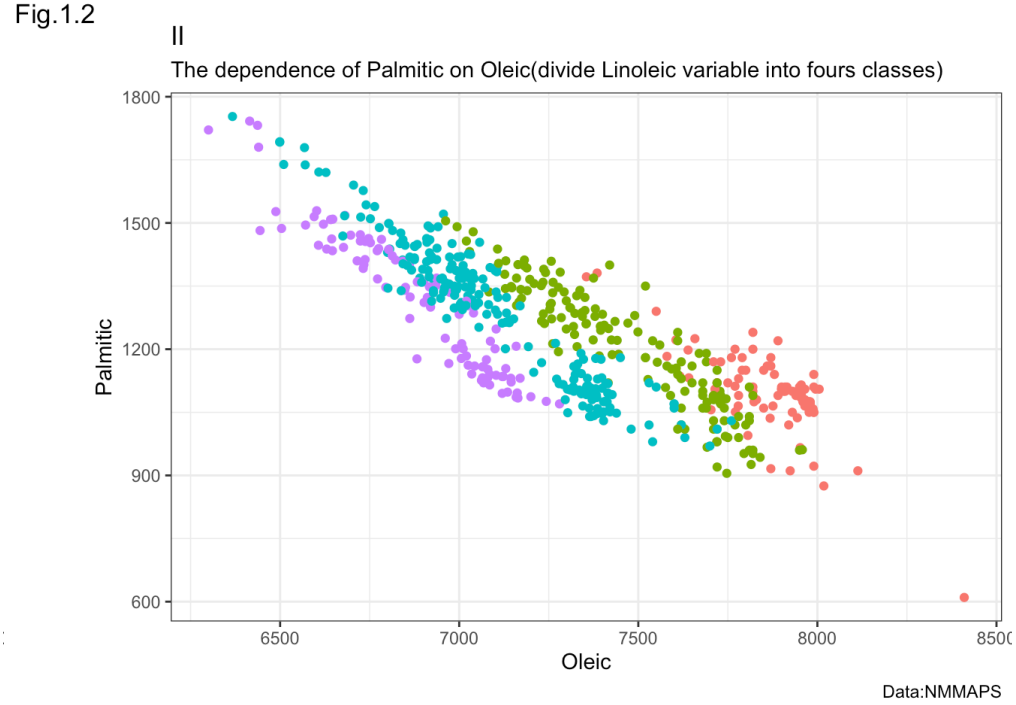


The second picture is more convenient for analysis. Compared with saturation, the human eye is more sensitive to changes in hue.We can perceive more levels of hue compare to that of saturation.

2. Task 2

```
p3<-ggplot()  +
  geom_point(data=x,aes(x=oleic,y=palmitic,size=discretized),color="blue",alpha=0.4)  +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="III",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.2.1") +
  theme(legend.position = 'none')

factor_data <- cut_interval(olive$linoleic, n = 4)

levels_data <- levels(factor_data)
available_angles <- seq(from=-pi, to=pi, length.out = 4)
angles <- c()
for (index in 1:length(olive$linoleic)) {
  item_level <- factor_data[index]
  if (item_level == levels_data[1]) {
    angles[index] <- available_angles[1]
  } else if (item_level == levels_data[2]) {
    angles[index] <- available_angles[2]
  } else if (item_level == levels_data[3]) {
    angles[index] <- available_angles[3]
  } else if (item_level == levels_data[4]) {
    angles[index] <- available_angles[4]
  } else {
    print("Something is Wrong")
    angles[index] <- 0
  }
}

p4 <- ggplot(olive, aes(x = oleic, y = palmitic)) +
  geom_point() +
  geom_spoke(aes(angle = angles, radius = 30))  +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="IV",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.2.2")

p2 + theme(legend.position = 'none')
```

Fig.1.2




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

Fig.2.1

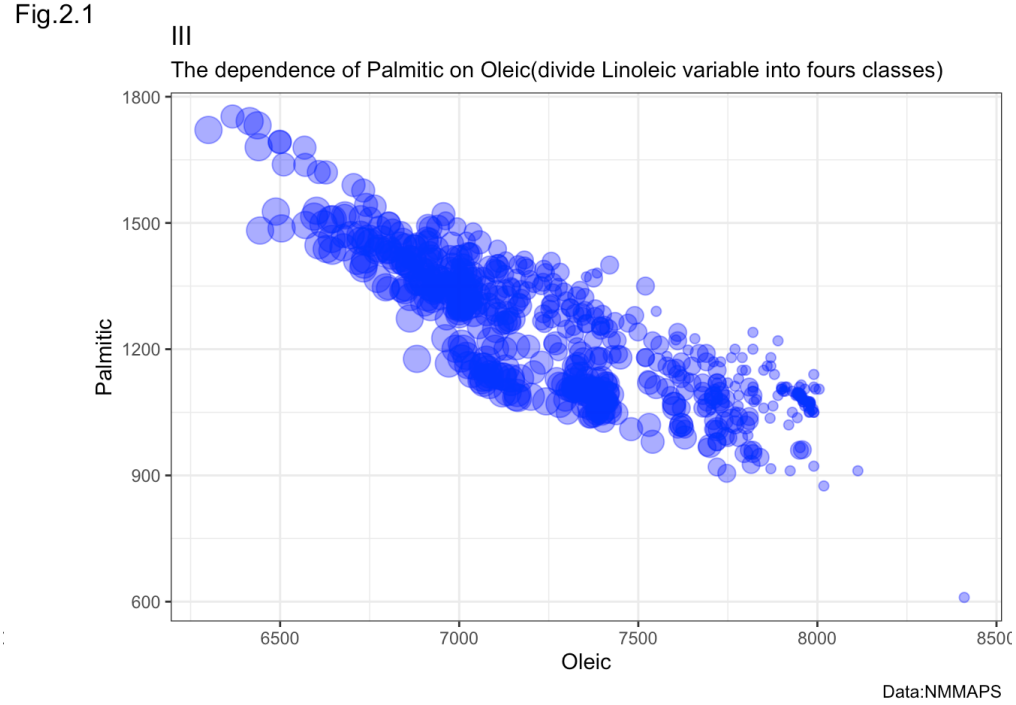
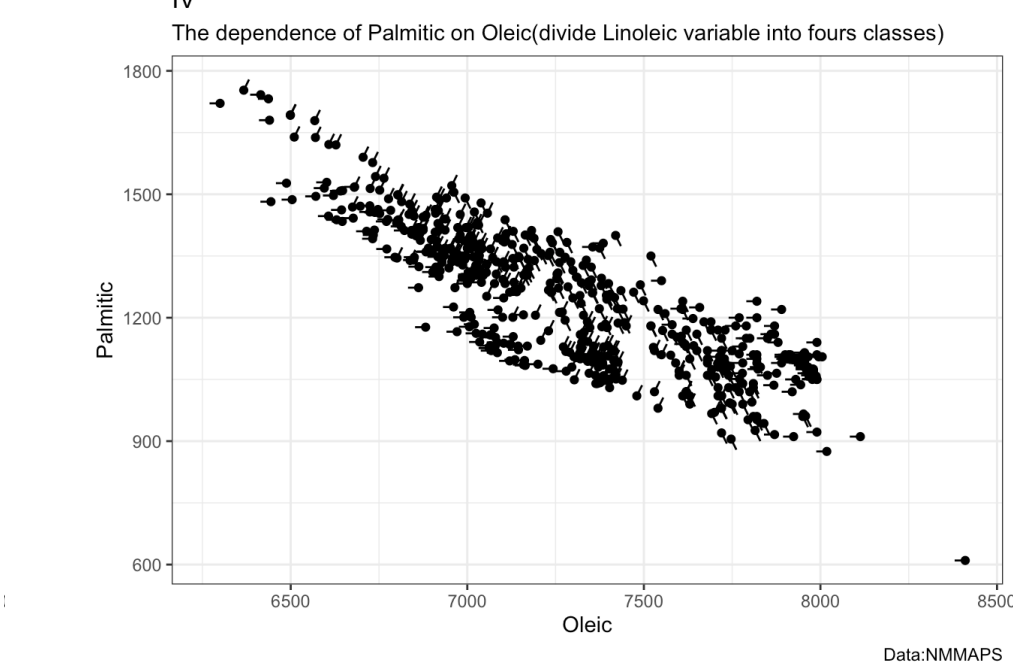


Fig.2.2



It's the most difficult to differentiate the category in the second plot(i.e. by size of the point),And it is easiest to differentiate the category by color.Consider of the perception metrics,the Hue feature has 3.1 bits of Channel capacity,meanwhile the Line orientation has 3 bit and the Size of squares has only 2.2 bits,which correspond the intuition.

3. Task 3

```
p5<-ggplot()  +
  geom_point(data=x,aes(x=eicosenoic,y=oleic,color=Region))+
  labs(x="Eicosenoic",y="Oleic",color="Region",
       title="V",
       subtitle="Oleic vs Eicosenoic(color is defined by numeric values of Region)",
       caption="Data:NMMAPS",
       tag="Fig.3.1")
y<-x
y$Region<-as.factor(y$Region)
p6<-ggplot()  +
  geom_point(data=y,aes(x=eicosenoic,y=oleic,color=Region))+
  labs(x="Eicosenoic",y="Oleic",color="Region",
       title="VI",
       subtitle="Oleic vs Eicosenoic(color is defined by categorical variable of Region)",
       caption="Data:NMMAPS",
       tag="Fig.3.2")
```

Fig.3.1

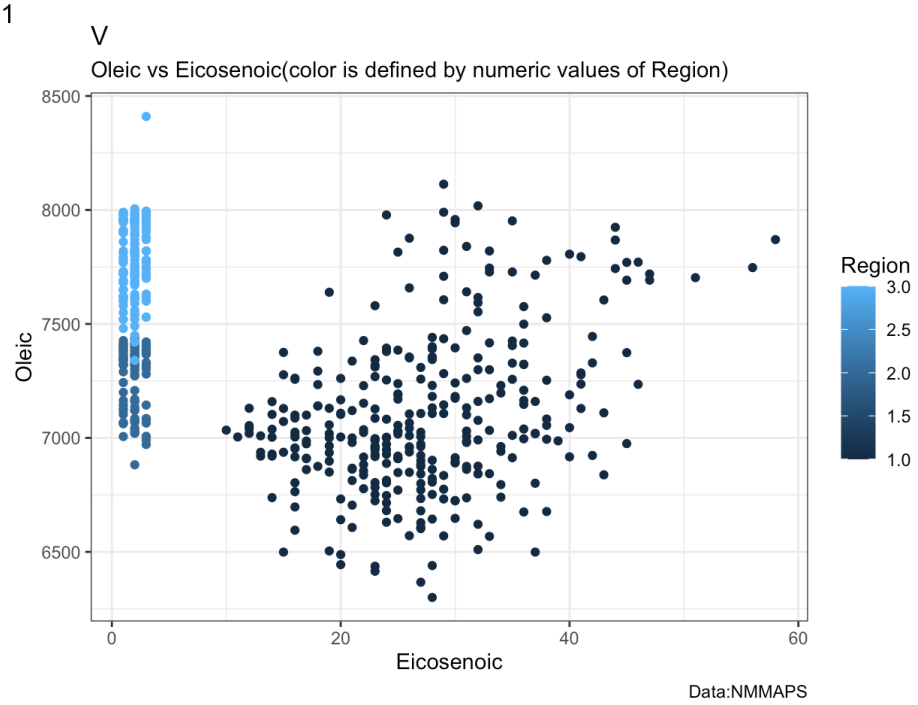
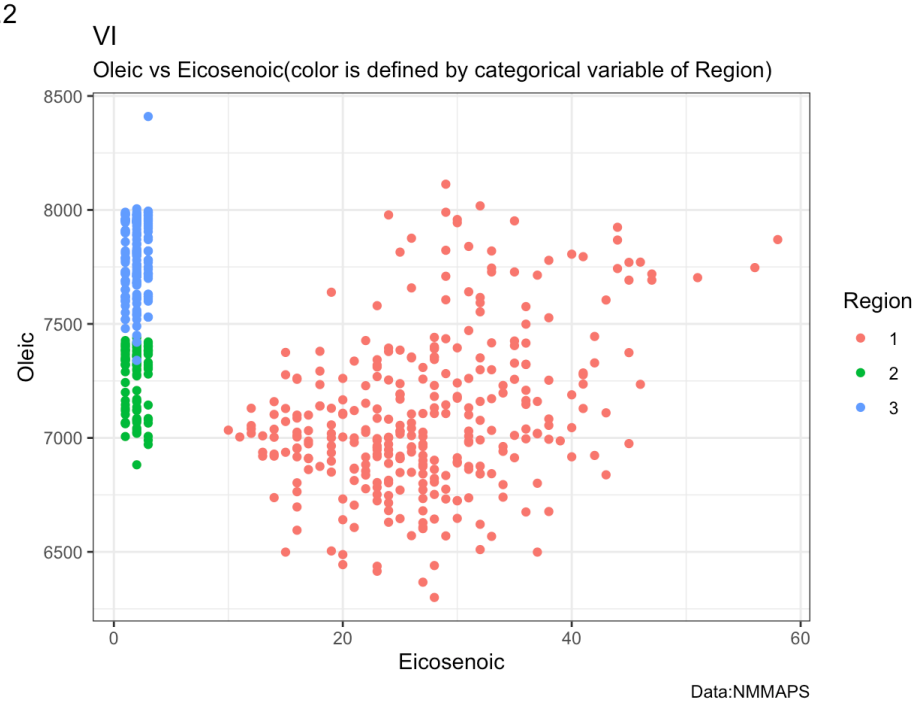


Fig.3.2



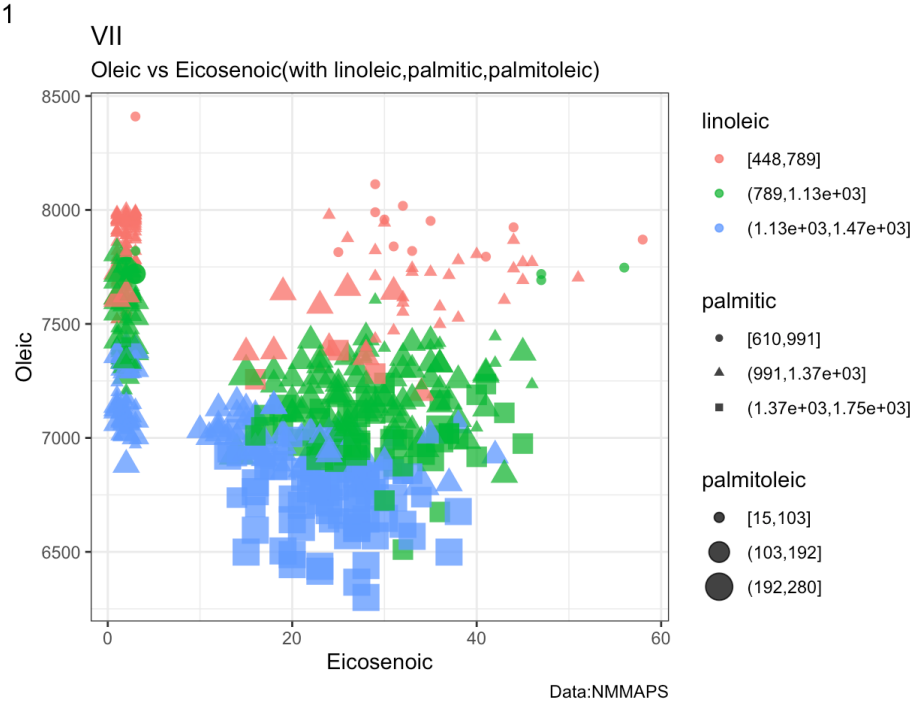
In the first picture, it is easy to distinguish the region of each sample point, but in the third picture, you can distinguish them faster.preattentive mechanism makes it possible.A unique visual property in the target makes it easy to be noticed,like Hue.But Value and saturation are not preattentive feature,so it will be slower.

4. Task 4

```
z<-data.frame(olive,discretized_linoleic=cut_interval(olive$linoleic, 3),
              discretized_palmitic=cut_interval(olive$palmitic, 3),
              discretized_palmitoleic=cut_interval(olive$palmitoleic, 3))
p7<-ggplot()  +
  geom_point(data=z,aes(x=eicosenoic,y=oleic,color=discretized_linoleic,
                        shape=discretized_palmitic,size=discretized_palmitoleic)
             ,alpha=0.8)+
  labs(x="Eicosenoic",y="Oleic",color="linoleic",
       shape="palmitic",
       size="palmitoleic",
       title="VII",
       subtitle="Oleic vs Eicosenoic(with linoleic,palmitic,palmitoleic)",
       caption="Data:NMMAPS",
       tag="Fig.4.1")
```

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

Fig.4.1



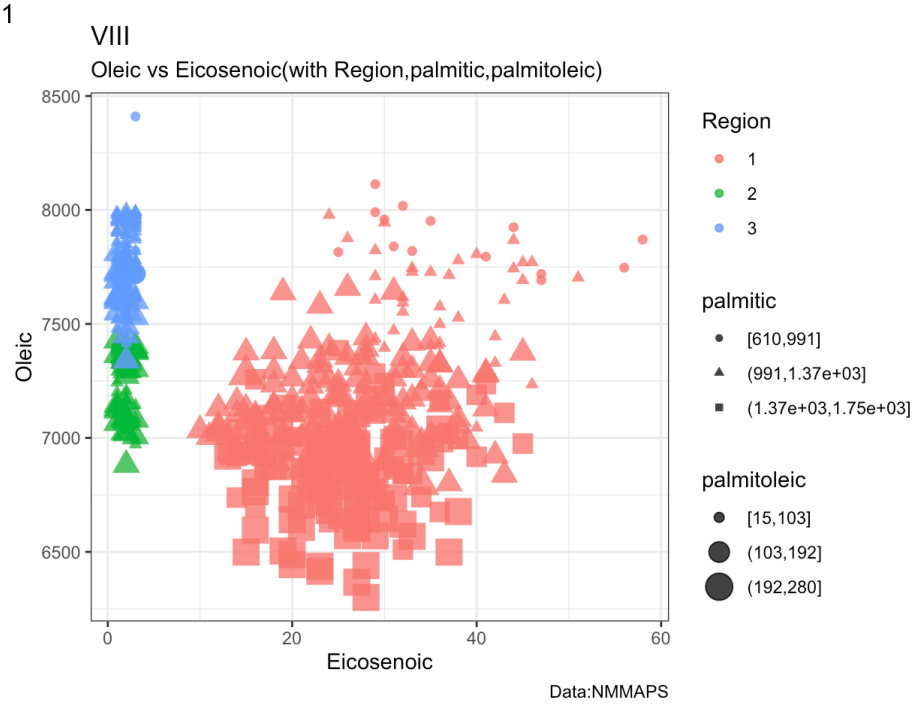
It's difficult to differentiate 27 types of observations,cause you need to compare 3 features between each point. Conjuction of features requires searial search between maps, it cannot be preattentively classified.

5. Task 5

```
z$Region<-as.factor(z$Region)
p8<-ggplot()  +
  geom_point(data=z,aes(x=eicosenoic,y=oleic,color=Region,
                        shape=discretized_palmitic,size=discretized_palmitoleic)
             ,alpha=0.8)+
  labs(x="Eicosenoic",y="Oleic",color="Region",
       shape="palmitic",
       size="palmitoleic",
       title="VIII",
       subtitle="Oleic vs Eicosenoic(with Region,palmitic,palmitoleic)",
       caption="Data:NMMAPS",
       tag="Fig.5.1")
```

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

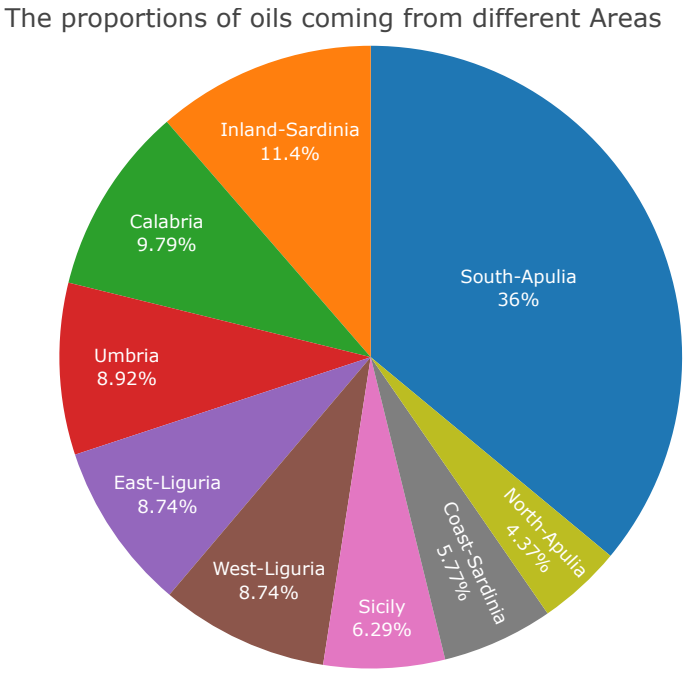
Fig.5.1



Why is it possible to clearly see a decision boundary between Regions despite many aesthetics are used? It's because the distribution of hues is very obvious, and human perceive hues in the preattention stage, it requires little effort or even realizes its occurrence and can only detect independent features.

6. Task 6

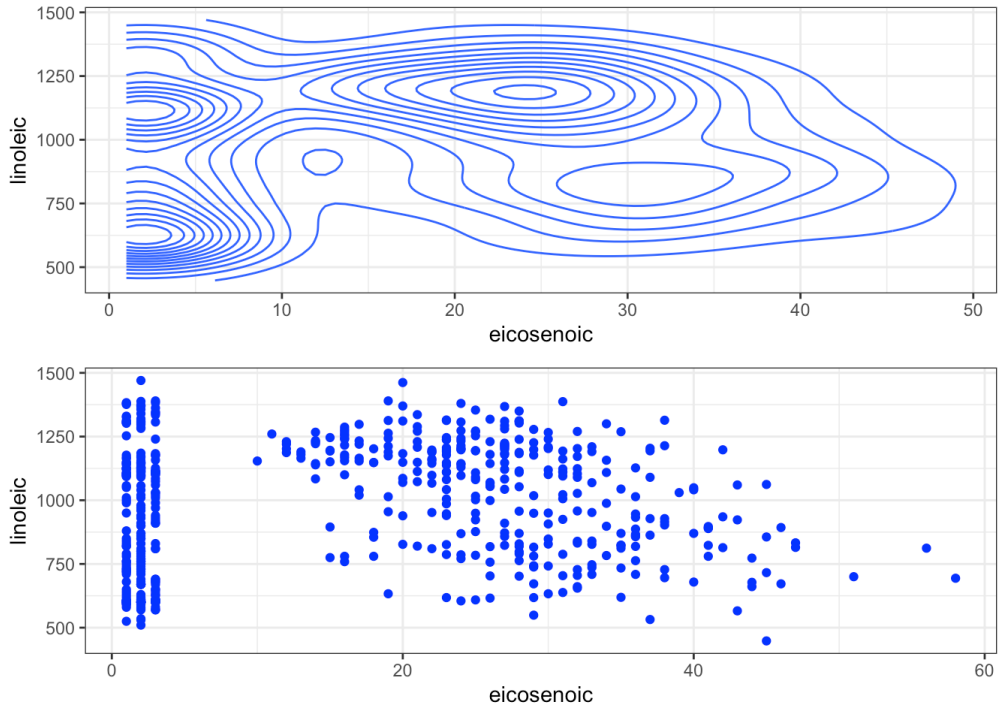
```
p9<-plot_ly(data=z,labels=~Area,type="pie",
            textposition = 'inside',
            textinfo = 'label+percent',
            insidetextfont = list(color = '#FFFFFF'))
p9<- p9 %>% layout(title = 'The proportions of oils coming from different Areas',
                   showlegend = F,
                   xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                   yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```



Pie charts are difficult to visualize the differences between data, but bar charts are just the opposite, and you can clearly see the differences between different data. Pie charts can't compare different values, and they can't convey more information.

7. Task 7

```
p10<-ggplot(z, aes(x =eicosenoic , y = linoleic)) +
  geom_density_2d()
p11<-ggplot(z, aes(x =eicosenoic , y = linoleic)) +
  geom_point(color="blue")
plotList2=list(p10,p11)
grid.arrange(grobs = plotList2)
```



The 2d-density contour plot have lines in an area eventhough there'are not sample points,which can be misleading.

Assignment 2

1. Load data

```
baseball <- read_xlsx("baseball-2016.xlsx")
```

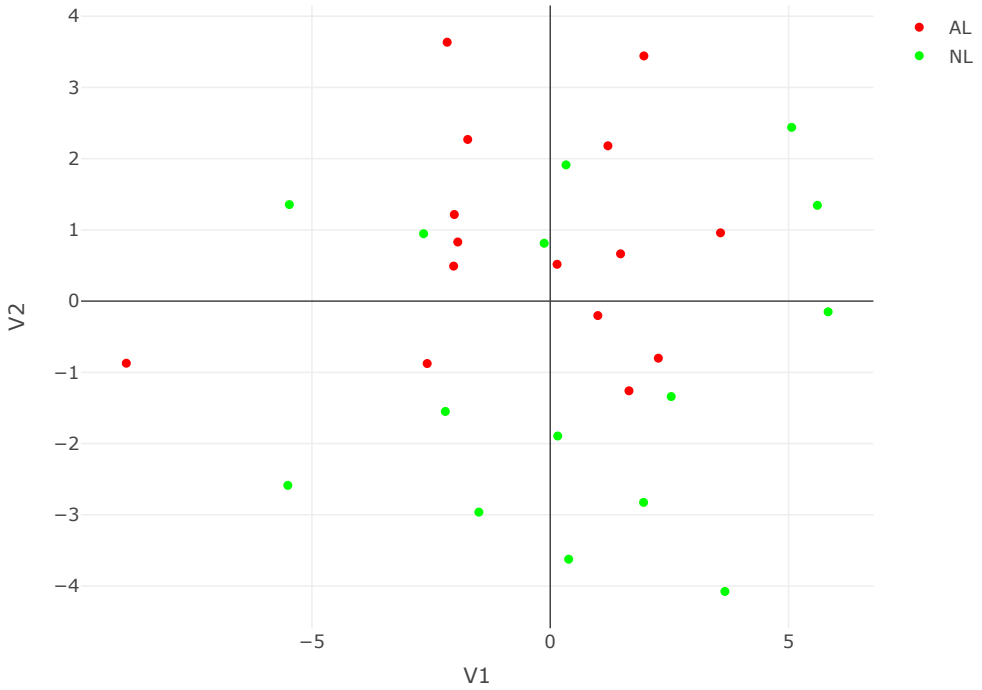
It is reasonable to scale the data in order to perform a MDS. The columns in this baseball data has different orders of magnitude. By directly using the original data, the column has a larger order of magnitude may will be highlighted and the column of the smaller one may will be weakened. The scaling will eliminate these differences and ensure the reliability of the results.

2. Non-metric MDS

```
distance <- dist(scale(baseball[,3:28]))
res <- isoMDS(distance, k = 2, p = 2)
```

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

```
coords <- res$points

baseball.mds <- as.data.frame(coords)
baseball.mds$team <- baseball$Team
baseball.mds$league <- baseball$League

plot_ly(baseball.mds,
        x = ~V1,
        y = ~V2,
        type = "scatter",
        hovertext = ~team,
        color = ~league,
        colors = c("red", "green"),
        mode = "markers")
```

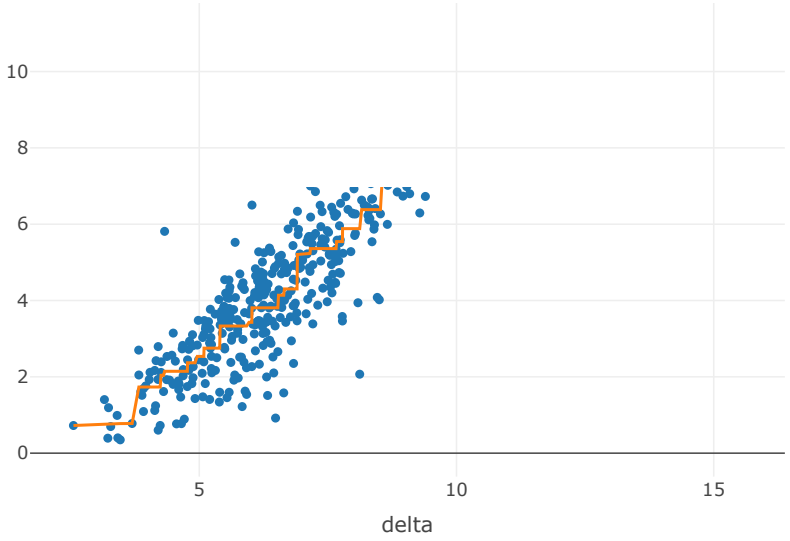


It looks the V2 dimension can separate the teams by leagues after no-metric MDS. Most AL teams' values on V2 axis are larger than zero and Most NL teams' values is smaller than zero.

Boston Res Sox, Chicago Cubs, Colorado Rockies and Atlanta Braves seem to be outliers among all the teams.

3. Shepard plot

```
shepard <- Shepard(distance, coords)
delta <- as.numeric(distance)
D <- as.numeric(dist(coords))

n <- nrow(coords)
index <- matrix(1:n, nrow=n, ncol=n)
obj1_index <- as.numeric(index[lower.tri(index)])
index <- matrix(1:n, nrow=n, ncol=n, byrow = T)
obj2_index <- as.numeric(index[lower.tri(index)])

teams <- baseball$Team
plot_ly() %>%
  add_markers(x=~delta, y=~D,
              hoverinfo = 'text',
              text = ~paste('Team1: ', teams[obj1_index], '<br> Team2: ', teams[obj2_index])) %>%
  add_lines(x=~shepard$x, y=~shepard$yf)
```

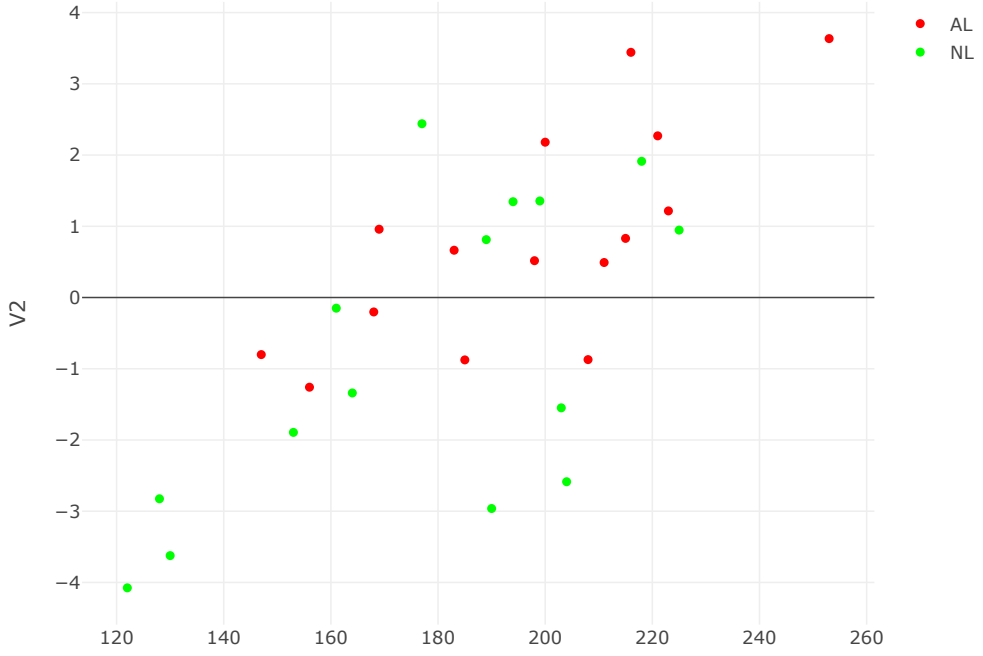


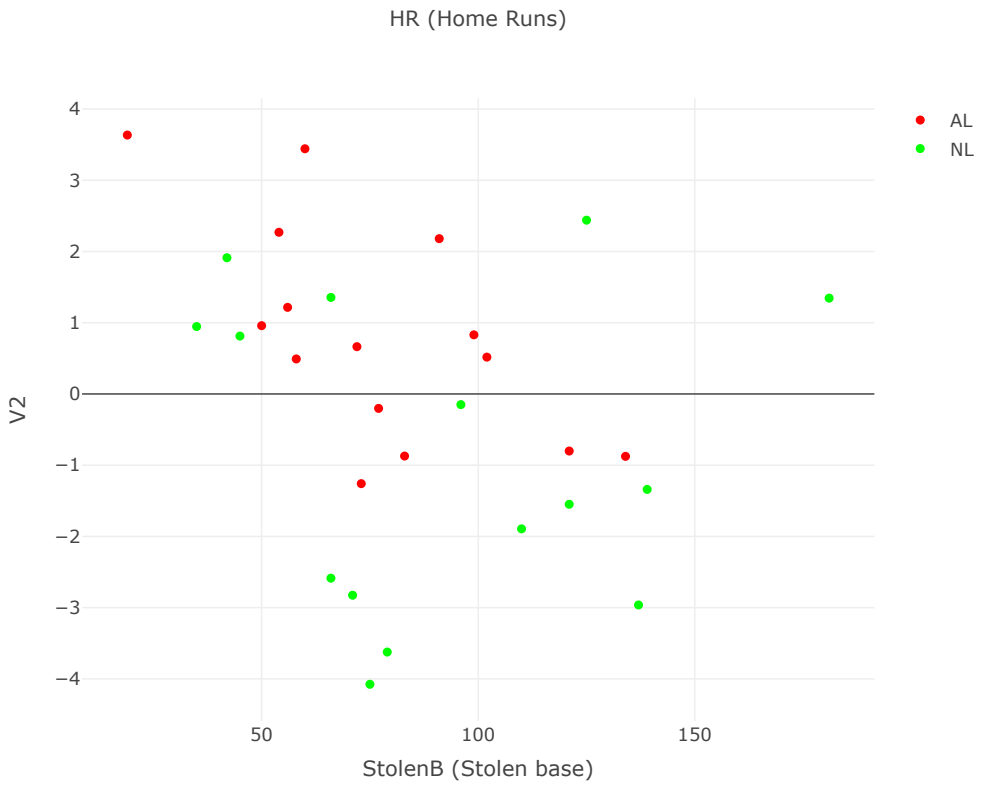
The scatter points are monotonically increasing overall, but there are some zero slopes in some intervals on trace 1. So We think this MDS is successful but not the best.

There are two observation pairs were hard for the MDS to map successfully: (Minnesota Twins, Aizona Diamondbacks) and (Oakland Athletics, Milwaukee Brewers).

4. Variables that can distinguish leagues

```
plot_mds_and_col <- function(mds_dims, col_index) {
  new_data <- data.frame(list(baseball[col_index], mds_dims))
  names(new_data) <- c("V1", "V2")
  new_data$team <- baseball$Team
  new_data$league <- baseball$League

  p <- plot_ly(new_data, x = ~V1, y = ~V2, type = "scatter",
          hovertext = ~team, color = ~league,
          colors = c("red", "green"), mode = "markers")
  return(p)
}

# HR (Home Runs)
p1 <- plot_mds_and_col(baseball.mds$V2, 12) %>%
  layout(xaxis = list(title = "HR (Home Runs)"))

# StolenB (Stolen base)
p2 <- plot_mds_and_col(baseball.mds$V2, 14) %>%
  layout(xaxis = list(title = "StolenB (Stolen base)"))
```





We think HR and StolenB appear to be two variables that can distinguish league between baseball teams among all scatter plots.

According the “MLB statistical standards” part of the Baseball statistics (https://en.wikipedia.org/wiki/Baseball_statistics) in Wikipedia, HR (Home Runs) and StolenB (Stolen base) are in the top 6 variables when value a baseball player in batting filed. We can also believe they are important for scoring a baseball team.

In fact, because of the differences in the rules of the two leagues, the games in AL are more power-based, which makes HR is more important in AL teams, while NL more emphasis on offense, and stealing bases is an method for offenses (Reference Link (http://www.differencebetween.net/miscellaneous/difference-between-al-and-nl/)). This can coincide with the results of our analysis.

Appendix

Codes For Assignment 1

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

theme_set(theme_bw())
olive=read.csv("olive.csv")

# Task 1
p1<- ggplot()  +
  geom_point(data=olive,aes(x=oleic,y=palmitic,color=linoleic)) +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="I",
       subtitle="The dependence of Palmitic on Oleic",
       caption="Data:NMMAPS",
       tag="Fig.1.1") +
  scale_colour_gradientn(colours = c("#66CCFF44", "#66CCFF"))
#Divide a continuous variable into three equal-sized groups
x<-data.frame(olive,discretized=cut_interval(olive$linoleic, 4))
p2<-ggplot()  +
  geom_point(data=x,aes(x=oleic,y=palmitic,color=discretized)) +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="II",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.1.2")

# Task 2
p3<-ggplot()  +
  geom_point(data=x,aes(x=oleic,y=palmitic,size=discretized),color="blue",alpha=0.4)  +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="III",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.2.1") +
  theme(legend.position = 'none')

factor_data <- cut_interval(olive$linoleic, n = 4)

levels_data <- levels(factor_data)
available_angles <- seq(from=-pi, to=pi, length.out = 4)
angles <- c()
for (index in 1:length(olive$linoleic)) {
  item_level <- factor_data[index]
  if (item_level == levels_data[1]) {
    angles[index] <- available_angles[1]
  } else if (item_level == levels_data[2]) {
    angles[index] <- available_angles[2]
  } else if (item_level == levels_data[3]) {
    angles[index] <- available_angles[3]
  } else if (item_level == levels_data[4]) {
    angles[index] <- available_angles[4]
  } else {
    print("Something is Wrong")
    angles[index] <- 0
  }
}

p4 <- ggplot(olive, aes(x = oleic, y = palmitic)) +
  geom_point() +
  geom_spoke(aes(angle = angles, radius = 30))  +
  labs(x="Oleic",y="Palmitic",color="Linoleic",
       title="IV",
       subtitle="The dependence of Palmitic on Oleic(divide Linoleic variable into fours classes)",
       caption="Data:NMMAPS",
       tag="Fig.2.2")

p2 + theme(legend.position = 'none')

# Task 3
p5<-ggplot()  +
  geom_point(data=x,aes(x=eicosenoic,y=oleic,color=Region))+
```

```
  labs(x="Eicosenoic",y="Oleic",color="Region",
       title="V",
       subtitle="Oleic vs Eicosenoic(color is defined by numeric values of Region)",
       caption="Data:NMMAPS",
       tag="Fig.3.1")
y<-x
y$Region<-as.factor(y$Region)
p6<-ggplot()  +
  geom_point(data=y,aes(x=eicosenoic,y=oleic,color=Region))+
  labs(x="Eicosenoic",y="Oleic",color="Region",
       title="VI",
       subtitle="Oleic vs Eicosenoic(color is defined by categorical variable of Region)",
       caption="Data:NMMAPS",
       tag="Fig.3.2")

# Task 4
z<-data.frame(olive,discretized_linoleic=cut_interval(olive$linoleic, 3),
              discretized_palmitic=cut_interval(olive$palmitic, 3),
              discretized_palmitoleic=cut_interval(olive$palmitoleic, 3))
p7<-ggplot()  +
  geom_point(data=z,aes(x=eicosenoic,y=oleic,color=discretized_linoleic,
                        shape=discretized_palmitic,size=discretized_palmitoleic)
             ,alpha=0.8)+
  labs(x="Eicosenoic",y="Oleic",color="linoleic",
       shape="palmitic",
       size="palmitoleic",
       title="VII",
       subtitle="Oleic vs Eicosenoic(with linoleic,palmitic,palmitoleic)",
       caption="Data:NMMAPS",
       tag="Fig.4.1")

# Task 5
z$Region<-as.factor(z$Region)
p8<-ggplot()  +
  geom_point(data=z,aes(x=eicosenoic,y=oleic,color=Region,
                        shape=discretized_palmitic,size=discretized_palmitoleic)
             ,alpha=0.8)+
  labs(x="Eicosenoic",y="Oleic",color="Region",
       shape="palmitic",
       size="palmitoleic",
       title="VIII",
       subtitle="Oleic vs Eicosenoic(with Region,palmitic,palmitoleic)",
       caption="Data:NMMAPS",
       tag="Fig.5.1")

# Task 6
p9<-plot_ly(data=z,labels=~Area,type="pie",
            textposition = 'inside',
            textinfo = 'label+percent',
            insidetextfont = list(color = '#FFFFFF'))
p9<- p9 %>% layout(title = 'The proportions of oils coming from different Areas',
                   showlegend = F,
                   xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                   yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

# Task 7
p10<-ggplot(z, aes(x =eicosenoic , y = linoleic)) +
  geom_density_2d()
p11<-ggplot(z, aes(x =eicosenoic , y = linoleic)) +
  geom_point(color="blue")
plotList2=list(p10,p11)
grid.arrange(grobs = plotList2)
```

Codes For Assignment 2

```r
library(readxl)
library(plotly)
library(MASS)

# Task 1
baseball <- read_xlsx("baseball-2016.xlsx")

# Task 2
distance <- dist(scale(baseball[,3:28]))
res <- isoMDS(distance, k = 2, p = 2)
coords <- res$points

baseball.mds <- as.data.frame(coords)
baseball.mds$team <- baseball$Team
baseball.mds$league <- baseball$League

plot_ly(baseball.mds,
        x = ~V1,
        y = ~V2,
        type = "scatter",
        hovertext = ~team,
        color = ~league,
        colors = c("red", "green"),
        mode = "markers")

# Task 3
shepard <- Shepard(distance, coords)
delta <- as.numeric(distance)
D <- as.numeric(dist(coords))

n <- nrow(coords)
index <- matrix(1:n, nrow=n, ncol=n)
obj1_index <- as.numeric(index[lower.tri(index)])
index <- matrix(1:n, nrow=n, ncol=n, byrow = T)
obj2_index <- as.numeric(index[lower.tri(index)])

teams <- baseball$Team
plot_ly() %>%
  add_markers(x=~delta, y=~D,
              hoverinfo = 'text',
              text = ~paste('Team1: ', teams[obj1_index], '<br> Team2: ', teams[obj2_index])) %>%
  add_lines(x=~shepard$x, y=~shepard$yf)

# Task 4
plot_mds_and_col <- function(mds_dims, col_index) {
  new_data <- data.frame(list(baseball[col_index], mds_dims))
  names(new_data) <- c("V1", "V2")
  new_data$team <- baseball$Team
  new_data$league <- baseball$League

  p <- plot_ly(new_data, x = ~V1, y = ~V2, type = "scatter",
          hovertext = ~team, color = ~league,
          colors = c("red", "green"), mode = "markers")
  return(p)
}

# HR (Home Runs)
plot_mds_and_col(baseball.mds$V2, 12) %>%
  layout(xaxis = list(title = "HR (Home Runs)"))

# StolenB (Stolen base)
plot_mds_and_col(baseball.mds$V2, 14) %>%
  layout(xaxis = list(title = "StolenB (Stolen base)"))
```