```
## Loading required packages
library(agricolae)
library(ggplot2)
library(ggpubr)
library(dplyr)
library(tidyverse)
library(corrplot)
library(xts)
library(dplyr)
library(PerformanceAnalytics)
library(zoo)
library(Formula)
library(readr)



### Read the csv file for Analysis
# Read csv file by R by giving another dataset name that is read by R

df1=read.csv(file.choose(), header = T)
df1

## To see the structure of the R read dataset df run the following command
str(df)
names(df)

# Variable conversion is needed before going to One-way ANOVA-DMRT test_SE-SD-Mean estimation

df$Replication = as.factor(df$Replication)
```

```
df$Treatment = as.factor(df$Treatment)
attach(df)
str(df)
names(df)

### One-way ANOVA_CRD Design with 3-Replicaitons_for the Parameter: Fruit Weight

FWanova <- aov(Fruitweight.g.~Treatment)
summary(FWanova)
### One-way ANOVA_CRD Design with 3-Replicaitons_for the Parameter: Fruit Diameter
FDManova <- aov(Fruitdiameter.cm. ~ Treatment)
summary(FDManova)
### One-way ANOVA_CRD Design with 3-Replicaitons_for the Parameter: Fruit Length
FLanova <- aov(Fruitlength.cm. ~ Treatment)
summary(FLanova)
### One-way ANOVA_CRD Design with 3-Replicaitons_for the Parameter: Yield
FYanova <- aov(Yieldper.plant..g. ~ Treatment)
summary(FYanova)

### Mean separation and lettering_Fruit Weight
# DMRT mean comparison test-for the parameter of Fruit Weight



## DMRT test for Fruit_Weight parameter (dependent variable)
### For Fruit Weight:

duncan.FW = duncan.test(y = Fruitweight.g.,
                        trt = Treatment,
```

```
                DFerror = FWanova$df.residual,
                MSerror = deviance(FWanova)/FWanova$df.residual,
                group = T,
                console = T)

## DMRT test for Fruit_Diameter parameter (dependent variable)
### For Fruit_Diameter:

duncan.FDM = duncan.test(y = Fruitdiameter.cm.,
                trt = Treatment,
                DFerror = FDManova$df.residual,
                MSerror = deviance(FDManova)/FDManova$df.residual,
                group = T,
                console = T)
## DMRT test for Fruit_Length parameter (dependent variable)
### For Fruit_Length:
duncan.FL = duncan.test(y = Fruitlength.cm.,
                trt = Treatment,
                DFerror = FLanova$df.residual,
                MSerror = deviance(FLanova)/FLanova$df.residual,
                group = T,
                console = T)


## DMRT test for Fruit_Yield parameter (dependent variable)
### For Fruit_Yield:
duncan.FY = duncan.test(y = Yieldper.plant..g.,
                trt = Treatment,
```

```
                DFerror = FYanova$df.residual,
                MSerror = deviance(FYanova)/FYanova$df.residual,
                group = T,
                console = T)
```

To see the SE for the parameter Fruit_Weight as per the independent variables of the studied (Treatment)

```
FW_SE = df %>%
  group_by(Treatment) %>%
  summarise(FW.mean = mean(Fruitweight.g.),
            std = sd(Fruitweight.g.),
            SE = sd(Fruitweight.g.)/sqrt(n()))

FW_SE
print(FW_SE)
```

To see the SE for the parameter Fruit_Diameter as per the independent variables of the studied (Treatment)

```
FDM_SE = df %>%
  group_by(Treatment) %>%
  summarise(FDM.mean = mean(Fruitdiameter.cm.),
            std = sd(Fruitdiameter.cm.),
            SE = sd(Fruitdiameter.cm.)/sqrt(n()))


print(FDM_SE)
```

To see the SE for the parameter Fruit_Length as per the independent variables of the studied (Treatment)

```
FL_SE = df %>%
  group_by(Treatment) %>%
  summarise(FL.mean = mean(Fruitlength.cm.),
            std = sd(Fruitlength.cm.),
            SE = sd(Fruitlength.cm.)/sqrt(n()))

print(FL_SE)
```

To see the SE for the parameter Fruit_Yield as per the independent variables of the studied (Treatment)

```
FY_SE = df %>%
  group_by(Treatment) %>%
  summarise(FY.mean = mean(Yieldper.plant..g.),
            std = sd(Yieldper.plant..g.),
            SE = sd(Yieldper.plant..g.)/sqrt(n()))

print(FL_SE)
```

```
#Correlation
names(df1)
#selecting only the numerical columns
numeric_vers <- df1%>%
  select(-Treatment,-Replication)%>%
  mutate_all(as.numeric)
```

```
numeric_vers


#compute the correlation matrix
cor_matrix <- cor(numeric_vers, use = "complete.obs", method = "pearson")
print(cor_matrix)


plot_cor_matrix <- corrplot(cor_matrix, method = "square",type = "full" ,diag = FALSE, tl.col =
"black", tl.srt = 45,tl.cex = .8,
                        title = "Correlation Matrix of Project Dataset", mar = c(1,.1,0.8,2))




#PCA
library(FactoMineR)
library(ggplot2)
library(mclust)
library(stats)
library(fpc)
library(ggforce)
library(grDevices)
library(factoextra)
library(stats)
#Now Doing PCA analysis on df
data.pca.df1 <- prcomp(df1[,-c(1:2)],
                       center = TRUE,
                       scale = TRUE)
data.pca.df1
p=data.pca.df1
summary(p)
```

```
df1$Treatment <- as.factor(df1$Treatment)
str(df1)


## To see the coordinates for the variables running res.var$coord command
res.var$coord

### To see the contributions of the each variable to the PCs running res.var$contrib
con <- res.var$contrib
# To see the quality of the representation of the variables running res.var$cos2
res.var$cos2
## To see the correlations between variables and dimensions running res.var$cor
res.var$cor
### Biplot Preparation using the Package: factoextra
library(factoextra)
fviz_pca_ind(p,col.ind="cos2",gradient.cols=c("#00AFBB","#E7B800","#FC4E07"),
             repel =   TRUE)
fviz_pca_ind(p,  col.ind="contrib",gradient.cols=c("#00AFBB","#E7B800","#FC4E07"),
             repel=TRUE)
## Variables_PCA-with Cos2 variable values_in circle shape with the variable contribution%-
cos2


fviz_pca_var(p,col.var="cos2",gradient.cols=c("#00AFBB","#E7B800","#FC4E07"),
             repel=TRUE)
### Variables_PCA-With the variables contribution_contrib to the each PCA components:
fviz_pca_var(p,            col.var="contrib",gradient.cols=c("#00AFBB","#E7B800","#FC4E07"),
             repel=TRUE)
fviz_pca_biplot(p,label="var", alpha.ind=1,col.var="blue",habillage=df$Treatment,
                repel=TRUE,addEllipses=FALSE, invisible="quali",legend.title="Treatment")+
```

```
  theme_bw()

df1_filtered <- df1  # Adjust this line based on how you filtered data for PCA



# Run PCA
p <- prcomp(df1_filtered[,-c(1:2)], center = TRUE, scale. = TRUE)

# Ensure the habillage factor matches the PCA dataset
df1_filtered$Treatment <- as.factor(df1_filtered$Treatment)
plot <- fviz_pca_biplot(p, axes = c(1, 2),
                        label = "var",                  # Labels for variables
                        geom.ind = "point",               # Individuals as points
                        col.var = "contrib",              # Color by contributions to the PC
                        gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  # Gradient colors for
variables
                        alpha.ind = 1,                  # Transparency for individuals
                        habillage = df1_filtered$Treatment, # Color individuals by Treatment
                        repel = TRUE,                   # Avoid text overlapping
                        addEllipses = FALSE,              # No confidence ellipses
                        legend.title = "Treatment",       # Title for the treatment legend
                        invisible = "quali",              # Hide qualitative supplementary variables
                        scale = "contrib")              # Scaling by contribution
+ coord_cartesian(xlim = c(-10, 10), ylim = c(-4, 4))  # Set axis limits
+ theme(text = element_text(size = 16),
        axis.text.x = element_text(color="black", size=16),
        axis.text.y = element_text(color="black", size=16),
        legend.position = "right")  # Position legend on the right side
```

```
+ labs(y = "PC2 (31.1%)", x = "PC1 (36.9%)", title = "PCA Biplot with Variable Contribution and Treatment Legend",
       color = "Variable Contribution")  # Labels for axes and color


# Print the plot
print(plot)
# Print the plot
print(plot + labs(y="PC2 (31.1%)", x = "PC1 (36.9%)"))


#dendo


install.packages("igraph")
library(igraph)


# Dendogram
df_mean <- df2 %>%
  group_by(Treatment)%>%  # Group data by Treatment
  summarise(across(everything(), mean, na.rm = TRUE))
# Changing Row names (generally row names are denoted by 1,2.... I want to denote the row as vegetable name so that when I will delete the 1st row of Veg name still I can see the vegetable )
df2 <- df1[,-1]
rownames(df_mean)<- c(df_mean$Treatment)
head(df_mean)
dim(df_mean)
df_mean$Treatment <- NULL


names(df_mean)
newdata <- (df[,3:23])
newdata
```

```
head(newdata)



#scale df_mean
df.scaled <- scale(df_mean)

res.dist <- dist(x = df.scaled,
                 method = "euclidean")
print(res.dist)

x <- as.matrix(res.dist)[1:11, 1:11]
x
round(x, digits = 3)
res.hc <- hclust(d = res.dist, method = "complete")  # Corrected from 'data' to 'd'

# Plot the result of the hierarchical clustering
plot(x = res.hc)



### Coloring
require(factoextra)
library(factoextra)
fviz_dend(x = res.hc, cex = 0.7, lwd = .7)
fviz_dend(x = res.hc, cex = 0.4, lwd = 0.8)
require(grDevices)
library(grDevices)
colors()
```

```
require(scales)

library(scales)

palette()

show_col(palette(rainbow(6)))

require("ggsci")

library(ggsci)

show_col(pal_jco(palette = c("default"))(10))

show_col(pal_jco("default", alpha = 0.6)(10))

# fviz_dend = Use fviz function for enhanced visualization of dendrogram

# x = an object of class dendrogram, hclust, agnes, diana, hcut, hkmeans or HCPC (FactoMineR).

# k = the number of groups for cutting the tree.

# cex = size of labels

# k_colors = a vector containing colors to be used for the groups.It should contains k number of colors. Allowed values include also "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; and scientific journal palettes from ggsci R package, e.g.: "npg", "aaas", "lancet", "jco", "ucscgb", "uchicago", "simpsons" and "rickandmorty"

library(NbClust)

fviz_nbclust(df_mean, kmeans, method= "gap_stat")


fviz_dend(x = res.hc, cex = 0.8, lwd = 0.8, k = 3,

        k_colors = c("red", "green3", "blue"))



fviz_dend(x = res.hc, cex = 0.8, lwd = 0.8, k = 3,

        k_colors = "jco")
```