Day 9 : Taxonomy in R

Ambu Vijayan

Bioinformatician

BioLit, Thiruvananthapuram

PCA

```
library(readr)
```

```
df <- read.csv("taxonomy_data.csv")
```

```
library(fixr)
```

```
df <- fix.data(df)
```

```
df$Phylum <- as.factor(df$Phylum)
```

Scatter Plot & Correlations

checking the correlation between independent variables

```
library(psych)
```

```
pairs.panels(df, gap = 0, bg = c("red", "yellow", "blue")[df$Phylum], pch=21)
```

Lower triangles provide scatter plots and upper triangles provide correlation values.

Principal Component Analysis

```
df.pca <- prcomp(df[,-1], center = TRUE,scale. = TRUE)
```

```
PCA <- df.pca
```

```
Summary_of_PCA <- summary(df.pca)
```

Standard deviation, Proportion of Variance and Cumulative Proportion

```
Summary_of_PCA
```

The first 5 principal components explain the variability around 94% and its captures the majority of the variability.

Proportion of Variance that first principal component explains 43% variance. Second component explains 22% variance. Third component explains 13% variance and so on.

```
df.pca.var <- df.pca$sdev ^ 2
```

```
propve <- df.pca.var / sum(df.pca.var)
```

Orthogonality

```
pairs.panels(df.pca$x, gap=0, bg = c("red", "yellow", "blue")[df.pca$Phylum], pch=21)
```

Now the correlation coefficients are zero, so we can get rid of multicollinearity issues.

Bi-Plot

```
library(ggbiplot)
```

```
g = ggbiplot(df.pca, obs.scale = 3, var.scale = 1, groups = df$Phylum, labels = row.names(data), circle = TRUE, circle.prob = 0.69, ellipse = FALSE)
```

```
g <- g + scale_color_discrete(name = '')
```

```
g <- g + theme(legend.direction = 'horizontal', legend.position = 'top')
```

```
Bi_Plot <- g
```

```
Bi_Plot
```

PC1 explains about 38% and PC2 explained about 27.5% of variability.

Arrows are closer to each other indicates the high correlation.

```
ggsave("ggbiplot_Alternanthera_micro.tiff", dpi=500, height=8, width=10, units="in")
```

Correlation matrix

```
com <- cor(df[,-1])
```

```
com
```

Higher values show positive correlation and Lower values show negetive correlation.

Computing eigen values and eigen vectors

```
df.pca.eigen <- eigen(cor(df[,-1]))
```

```
eigen values
```

```
df.pca.eigen$values
```

```
eigen vectors
```

```
df.pca.eigen$vectors
```

A scree plot is used to access components or factors which explains the most of variability in the data. It represents values in descending order.

Plot for the variance explained for each principal component

```
plot(propve, xlab = "principal component", ylab = "Proportion of Variance Explained", ylim = c(0, 1), type = "b", main = "Scree Plot")
```

Plot for the cumulative proportion of variance explained

```
plot(cumsum(propve), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained", ylim = c(0, 1), type = "b")
```

Plot for the First 10 Principal Components against Eigen Value and Percentage of explained variances.

```
library(factoextra)
```

```
fviz_eig(df.pca, addlabels = TRUE, xlab = "Principal Component")
```

```
fviz_eig(df.pca, choice = "eigenvalue", addlabels = TRUE, xlab = "Principal Component")
```

```
res.eig <- df.pca.eigen$values
```

```
res.eig <- round((df.pca.eigen$values),2)
```

```
res.eig <- data.frame(t(res.eig))
```

```
rownames(res.eig) <- ("Eigen_Values")
```

```
result <- df.pca$rotation
```

```
colnames(res.eig) <- (colnames(result))
```

```
comp.pca <- Summary_of_PCA$importance
```

```
comp.pca <- data.frame(comp.pca)
```

```
rownames(comp.pca) <- c('Standard deviation', 'Percentage of Variance',
'Cumulative Percentage')
```

```
comp.sd <- comp.pca[1,]
```

```
comp.pv <- comp.pca[2,]*100
```

```
comp.cp <- comp.pca[3,]*100
```

```
comp.pca <- rbind(comp.sd,comp.pv,comp.cp)
```

```
result <- round((df.pca$rotation),2)
```

```
result <- rbind(result,res.eig,comp.pca)
```

```
write.csv(result, "Alter_micro_result.csv", row.names = TRUE)
```

```
library(ggplot2)

library(readr)

library(dplyr)

library(dendextend)

df <- read.csv("taxonomy_data.csv")

df.group <- split(df[,2:7], df$Phylum)

df.means <- sapply(df.group, function(x) { apply(x, 2, mean)}, simplify = 'data.frame')

df.means.t <- t(df.means)

Mean_Data <- df.means.t
```

Means of Data

```
Mean_Data
```

```
d <- dist(df.means.t, method = "euclidean")
```

```
Calculated_Distance <- d
```

```
hc1 <- hclust(d, method = "average" )
```

```
Cluster_Data <- hc1
```

Distance Calculation

Calculated_Distance

```
plot(hc1, cex = 0.6, hang = -1)
```

```
par(mar=c(2,2,6,2))
```

```
plot(hc1, cex = 0.6)
```

```
rect.hclust(hc1, k = 5, border = 2:5)
```

```
abline(h = 15, col = 'red')
```

Performing ANOVA

```
APA.aov <- anova(lm(APA ~ Phylum, data = df))
```

```
KLA.aov <- anova(lm(KLA ~ Phylum, data = df))
```

```
Anova_for_APA <- APA.aov
```

```
Anova_for_KLA <- KLA.aov
```

```
Anova_for_APA
```

```
Anova_for_KLA
```

Summary of ANOVA

```
Summary_of_Anova_for_APA <- summary(APA.aov)
```

```
Summary_of_Anova_for_KLA <- summary(KLA.aov)
```

```
Summary_of_Anova_for_APA
```

```
Summary_of_Anova_for_KLA
```

Plotting the Graph for ANOVA

```
par(mar=c(2, 2, 2, 2))
```

```
plot(Anova_for_APA)
```

```
mtext("APA", side = 3, line = 1)
```

```
plot(Anova_for_KLA)
```

```
mtext("KLA", side = 3, line = 1)
```

ANOVA using another package for confirmation

```
aov_APA <- aov(APA ~ Phylum, data = df)
```

```
aov_KLA <- aov(KLA ~ Phylum, data = df)
```

```
Summary_of_Anova_for_APA <- summary(aov_APA)
```

```
Summary_of_Anova_for_KLA <- summary(aov_KLA)
```

```
Summary_of_Anova_for_APA
```

```
Summary_of_Anova_for_KLA
```

Plotting Density

```
library(ggplot2)
```

```
ggplot(df, aes(APA)) +
geom_density(aes(APA, fill = Phylum), position = 'identity', alpha = 0.5) +
labs(x = 'APA', y = 'Density') + scale_fill_discrete(name = 'Phylum') +
ggtitle("Density plot of APA")
```

```
ggplot(df, aes(KLA)) +
geom_density(aes(KLA, fill = Phylum), position = 'identity', alpha = 0.5) +
labs(x = 'KLA', y = 'Density') + scale_fill_discrete(name = 'Phylum') +
ggtitle("Density plot of KLA")
```

Calculating MEAN values individually

```
means_APA <- round(tapply(df$APA, df$Phylum, mean), digits=2)
```

```
means_KLA <- round(tapply(df$KLA, df$Phylum, mean), digits=2)
```

```
Mean_Value_for_APA <- means_APA
```

```
Mean_Value_for_KLA <- means_KLA
```

```
Mean_Value_for_APA
```

```
Mean_Value_for_KLA
```

Combined MEAN

```
df.group <- split(df[,2:7], df$Phylum)
```

```
df.means <- sapply(df.group, function(x) { apply(x, 2, mean) }, simplify = 'data.frame')
```

```
df.means.t <- t(df.means)
```

```
Combined_Mean <- df.means.t
```

```
Combined_Mean
```

Post hoc testing

```
Tuckey_aov_APA <- TukeyHSD(aov_APA, conf.level=.95)
```

```
Tuckey_aov_KLA <- TukeyHSD(aov_KLA, conf.level=.95)
```

```
Tuckey_for_APA <- Tuckey_aov_APA
```

```
Tuckey_for_KLA <- Tuckey_aov_KLA
```

```
Tuckey_for_APA
```

```
Tuckey_for_KLA
```

Plotting 95% Confidence level for Tuckey

```
par(mar=c(2, 15, 4, 2))
```

```
plot(Tuckey_aov_APA, las = 2, cex.axis=0.6)
mtext("number of Pores", side = 3, line = 1)
```

```
plot(Tuckey_aov_KLA, las = 2, cex.axis=0.6)
mtext("KLA", side = 3, line = 1)
```