Day 8 : Taxonomy in R

Ambu Vijayan

Bioinformatician

BioLit, Thiruvananthapuram

Various analyses on the provided taxonomy count data.

1. Descriptive Statistics
2. Data Visualization
3. Diversity Analysis
4. Comparative Analysis
5. Correlation Analysis
6. Clustering
7. Principal Component Analysis (PCA)
8. ANOVA or Kruskal-Wallis Test
9. Regression Analysis

Import Data

```
taxonomy_data <- read_excel("D:/Users/ambuv/Downloads/Genomic data for Bioinfo class.xlsx")
```

Export Data

```
write.csv(taxonomy_data, "taxonomy_data.csv", row.names = F)
```

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

View Data

```
View(taxonomy_data)
```

Overall Summary

```
summary(taxonomy_data)
```

Individual Summary

```
library(dplyr)
```

Group by Phylum and calculate summary statistics

```
summary_stats <- taxonomy_data %>%   group_by(Phylum) %>%   summarise(     Mean = mean(APA),     Median = median(APA),     Min = min(APA),     Max = max(APA)   )
```

View the summary statistics

```
print(summary_stats)
```

Data Visualization

Install and load necessary packages

```
library(ggplot2)
```

Create a bar plot for the APA sample

```
ggplot(taxonomy_data, aes(x = Phylum, y = APA, fill = Phylum)) +   geom_bar(stat
= "identity") +   labs(title = "Abundance of Phylum in APA Sample", x = "Phylum",
y = "Abundance") +   theme_minimal()
```

Changing angle of axis legends

```
ggplot(taxonomy_data, aes(x = Phylum, y = APA, fill = Phylum)) +   geom_bar(stat
= "identity") +   labs(title = "Abundance of Phylum in APA Sample", x = "Phylum",
y = "Abundance") +   theme_minimal() +   theme(axis.text.x = element_text(angle =
45, hjust = 1))
```

Heatmap

Install and load necessary packages

```
install.packages("gplots")
```

```
library(gplots)
```

```
heatmap.2(as.matrix(taxonomy_data[, 2:7]),            Rowv = TRUE, Colv = TRUE,
        dendrogram = 'row',           col = heat.colors(256),           main =
"Heatmap of Phylum Abundance in Different Samples")
```

Diversity Analysis

Install and load necessary packages

```
install.packages("vegan")
library(vegan)
```

```
abundance_data <- taxonomy_data[, 2:7]
```

Calculate Shannon diversity index for each sample

```
shannon_index <- diversity(abundance_data, index = "shannon")
```

Print the Shannon diversity indices

```
print(shannon_index)
```

Calculate Simpson's diversity index for each sample

```
simpson_index <- diversity(abundance_data, index = "simpson")
```

Print the Simpson's diversity indices

```
print(simpson_index)
```

```
diversity_results <- data.frame(
Sample = taxonomy_data$Phylum,
Shannon_Index = shannon_index,
Simpson_Index = simpson_index
)
```

Print the diversity results

```
print(diversity_results)
```

Install and load necessary packages

```
library(ggplot2)
```

Create a bar plot for Shannon diversity index

```
ggplot(diversity_results, aes(x = Sample, y = Shannon_Index)) +
geom_bar(stat = "identity", fill = "blue") +
labs(title = "Shannon Diversity Index for Each Sample", x = "Sample", y = "Shannon Diversity Index") +
theme_minimal()
```

Create a bar plot for Simpson's diversity index

```
ggplot(diversity_results, aes(x = Sample, y = Simpson_Index)) +
geom_bar(stat = "identity", fill = "green") +
labs(title = "Simpson's Diversity Index for Each Sample", x = "Sample", y = "Simpson's Diversity Index") +
theme_minimal()
```

Comparative Analysis

Stack the data into long format for ANOVA

```
library(tidyr)
long_data <- gather(taxonomy_data, key = "Sample", value = "Abundance", -Phylum)
```

Perform ANOVA

```
anova_result <- aov(Abundance ~ Phylum, data = long_data)
```

Print the ANOVA result

```
print(summary(anova_result))
```

Correlation Analysis

Select relevant columns for correlation analysis (e.g., APA to VKM columns)

```
selected_data <- taxonomy_data[, 2:7]
```

Calculate the correlation matrix

```
correlation_matrix <- cor(selected_data)
```

Print the correlation matrix

```
print(correlation_matrix)
```

Install and load necessary packages

```
install.packages("corrplot")
library(corrplot)
```

Create a heatmap of the correlation matrix

```
corrplot(correlation_matrix, method = "color", type = "upper", addCoef.col = "black",
tl.col = "black", tl.srt = 45)
```

Clustering

Select relevant columns for clustering (e.g., APA to VKM columns)

```
selected_data <- taxonomy_data[, 2:7]
```

Calculate Euclidean distances

```
distances <- dist(selected_data)
```

Perform hierarchical clustering

```
hierarchical_cluster <- hclust(distances)
```

Plot the dendrogram

```
plot(hierarchical_cluster, main = "Hierarchical Clustering Dendrogram", xlab = "Samples", sub = NULL)
```

Cut the dendrogram to form clusters

```
clusters <- cutree(hierarchical_cluster, k = 3)
```

Adjust the number of clusters as needed

Add cluster information to the original data

```
taxonomy_data$Cluster <- clusters
```

Print the cluster assignments

```
print(taxonomy_data[, c("Phylum", "Cluster")])
```

You can also use k-means clustering

Select relevant columns for clustering (e.g., APA to VKM columns)

```
selected_data <- taxonomy_data[, 2:7]
```

Perform k-means clustering (adjust the number of clusters as needed)

```
kmeans_cluster <- kmeans(selected_data, centers = 3, nstart = 10)
```

Add cluster information to the original data

taxonomy_data$Cluster < -kmeans_{c}luster$cluster

Print the cluster assignments

print(taxonomy_data[, c("Phylum", "Cluster")])

Principal Component Analysis (PCA)

Perform PCA

```
pca_result <- prcomp(taxonomy_data[, 2:7], scale. = TRUE)
```

Get the principal components

```
pcs <- pca_result$x
```

```
taxonomy_data$Phylum)
```

Plot using ggplot2

```
ggplot(plot_data, aes(x = PC1, y = PC2, color = Phylum)) +
geom_point() +
labs(title = "PCA: First Two Principal Components", x = "PC1", y = "PC2") +
theme_minimal()
```

Plot using ggplot2 with labels

```
ggplot(plot_data, aes(x = PC1, y = PC2, color = Phylum, label = Phylum)) +
geom_point() +
geom_text(hjust = 0, vjust = 0, check_overlap = TRUE) + # Adjust text position
labs(title = "PCA: First Two Principal Components", x = "PC1", y = "PC2") +
theme_minimal()
```

Perform simple linear regression

```
regression_model <- lm(selected_data[, 1] ~ selected_data[, 2])
```

Print the regression summary

```
print(summary(regression_model))
```

Visualise and Calculate R squared

Install and load necessary packages

```
library(ggplot2)
```

Perform simple linear regression

```
regression_model <- lm(selected_data[, 1] ~ selected_data[, 2])
```

Create a data frame for plotting

```
plot_data <- data.frame(APA = selected_data[, 1], KLA = selected_data[, 2])
```

Plot the data points

```
ggplot(plot_data, aes(x = KLA, y = APA)) +   geom_point() +   labs(title =
"Linear Regression: APA vs. KLA", x = "KLA", y = "APA") +   theme_minimal() +
geom_smooth(method = "lm", se = FALSE, color = "blue") +   annotate("text", x =
max(plot_data$KLA), y = max(plot_data$APA),            label = paste("Y =",
round(coef(regression_model)[1], 3), "+",
round(coef(regression_model)[2], 3), "* X"),            hjust = 1, vjust = 1,
color = "red") +   annotate("text", x = max(plot_data$KLA), y =
max(plot_data$APA) - 200,            label = paste("R-squared =",
round(summary(regression_model)$r.squared, 3)),            hjust = 1, vjust = 1,
color = "red")
```

1. **`ggplot(plot_data, aes(x = KLA, y = APA))`**:
   - `ggplot` initializes a ggplot object with the specified data (`plot_data`) and aesthetic mappings (`aes`).
   - `x = KLA` specifies that the x-axis should be based on the 'KLA' column of the data.
   - `y = APA` specifies that the y-axis should be based on the 'APA' column of the data.
2. **`geom_point()`**:
   - `geom_point` adds a layer to the plot for scatter points, creating a scatterplot of the data.

3. **`labs(title = "Linear Regression: APA vs. KLA", x = "KLA", y = "APA")`**:
    - `labs` is used to set the title and axis labels of the plot.
4. **`theme_minimal()`**:
    - `theme_minimal` sets the visual theme of the plot to a minimal style.
5. **`geom_smooth(method = "lm", se = FALSE, color = "blue")`**:
    - `geom_smooth` adds a smoothed line to the plot. Here, `method = "lm"` specifies that the line should be a linear regression line, and `se = FALSE` suppresses the display of confidence intervals around the line. The line is colored blue.

6. **`annotate("text", x = max(plot_data$KLA), y = max(plot_data$APA), ...)` and `annotate("text", x = max(plot_data$KLA), y = max(plot_data$APA) - 200, ...)`:**
    - `annotate("text", ...)` adds text annotations to the plot.
    - The first `annotate` call adds the equation of the regression line.
    - The second `annotate` call adds the R-squared value.
    - `x = max(plot_data$KLA)` and `y = max(plot_data$APA)` position the text annotations at the top-right corner of the plot.
    - `hjust = 1, vjust = 1` adjust the justification of the text.