Detailed Explanation of the R Tutorial

1 Introduction

This tutorial explores the basics of R programming by utilizing the iris dataset. It covers installing necessary packages, data manipulation using `dplyr`, and data visualization using base R graphics and `ggplot2`.

2 Installing and Loading Packages

To begin, we need to install and load the `ggplot2` and `dplyr` packages. These packages provide powerful tools for data visualization and data manipulation respectively.

```
install.packages("ggplot2")
library(ggplot2)

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

When loading these packages, you might see some messages regarding masking of functions. This is normal and indicates that functions from `dplyr` are taking precedence over those in base R.

3 Loading the Dataset

The iris dataset is included in R and can be loaded with the `data` function. This dataset contains measurements of iris flowers and is commonly used for demonstrating data analysis techniques.

```
data(iris)
print(head(iris))
```

The `head` function displays the first six rows of the dataset, allowing you to quickly inspect its structure.

4 Data Manipulation with `dplyr`

The `dplyr` package offers a range of functions to manipulate data frames in R. Below, we use the `select` function to extract specific columns from the iris dataset.

```
selectedColumns <- select(iris, Sepal.Length, Petal.Length)
print(head(selectedColumns))
```

The `select` function is used to pick columns by their names, making it easy to create a subset of the data.

5 Data Visualization

R provides various functions to visualize data. We start with a histogram to show the distribution of sepal lengths in the iris dataset. In large amounts of biological data, histograms can be used to demonstrate basic relationships and distributions of data.

5.1 Histogram

```
hist(iris$Sepal.Length,
     col = 'orange',
     main = 'Histogram',
     xlab = 'Sepal Length',
     ylab = 'Frequency')
```

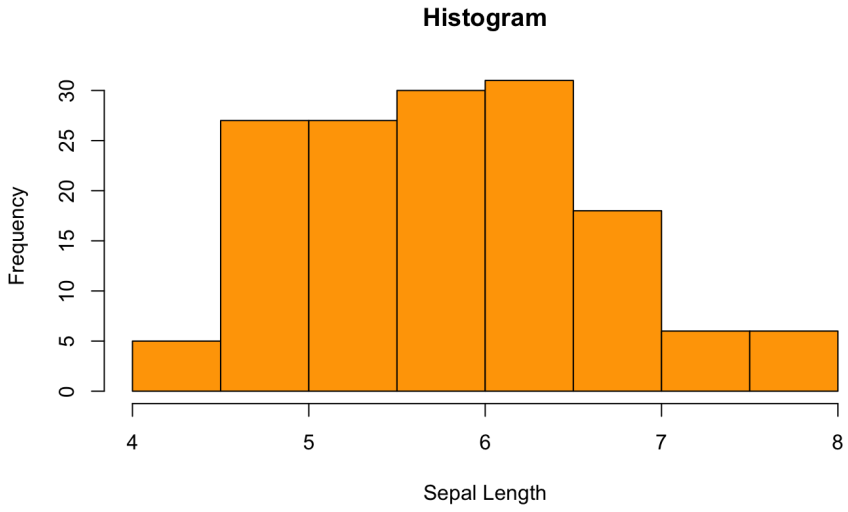

Figure 1: An example histogram produced by the code above.

The `hist` function creates a histogram. Here, we specify the color, title, and labels for the x and y axes. Note that the function specifically calls the `Sepal.Length` variable using the `$`.

5.2 Scatter Plot

A scatter plot can be used to explore the relationship between petal length and petal width. A scatter plot can be used to demonstrate important relationships in large amounts of data - a critical part of computational biology.

```
plot(iris$Petal.Length, iris$Petal.Width,
     col = 'orange',
     main = 'Scatter Plot',
     xlab = 'Petal Length',
     ylab = 'Petal Width',
     pch = 15)
```

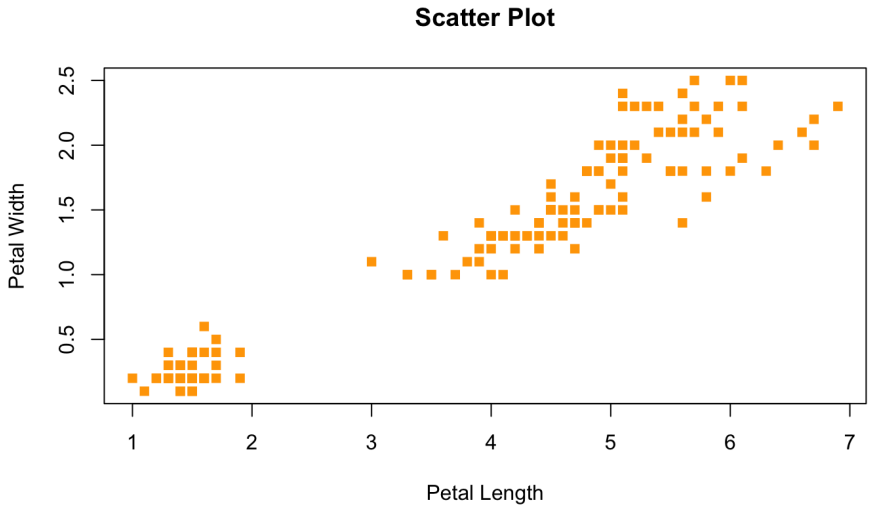

Figure 2: An example scatter plot produced by the code above.

The `plot` function creates a scatter plot, where `$` specifies the plotting variable.

5.3 Box Plot

Box plots are useful for comparing distributions across categories. Here, we compare the petal width across different species of iris. Even in large datasets, box

plots can be useful, especially for understanding central tendency and spread. They should typically be used in early stages of data analysis.

```
boxplot(Petal.Width~Species,
        data = iris,
        main = 'Petal Width with regards to Species',
        xlab = 'Species',
        ylab = 'Petal Width',
        col = 'orange')
```

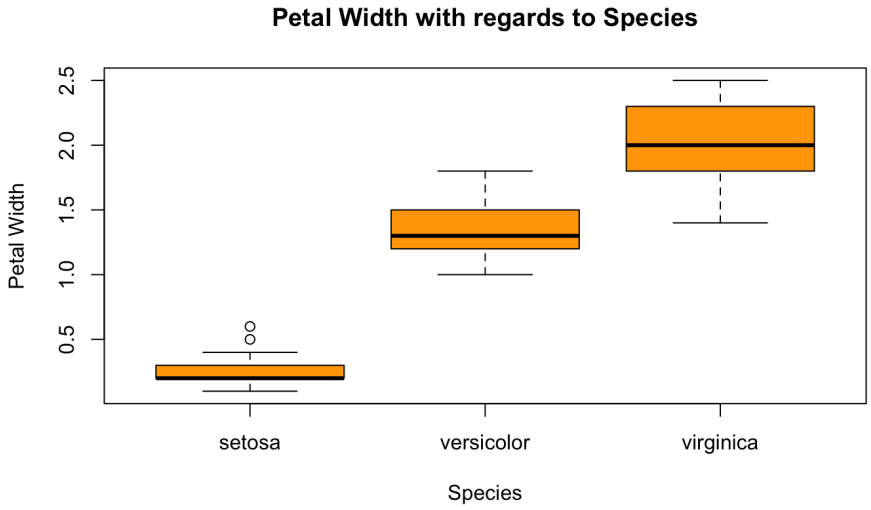

Figure 3: An example box plot produced by the code above. Note that the open dots represent outliers in the figure.

The `boxplot` function generates a box plot, showing the spread and center of the data for each species. Note that the text immediately surrounding the ~ specifies the x-axis and the y-axis.

6 Conclusion

This tutorial provides an introduction to R programming, data manipulation with `dplyr`, and data visualization with base R functions. The iris dataset serves as a simple yet comprehensive example to demonstrate these techniques.

To practice further, try creating similar visualizations with other datasets available in R. Use the `data()` function to see a list of all available datasets.