Examination

Nishan Regmi

2024-05-31

```r
#Question no 6
##part (a)
# Load necessary libraries
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

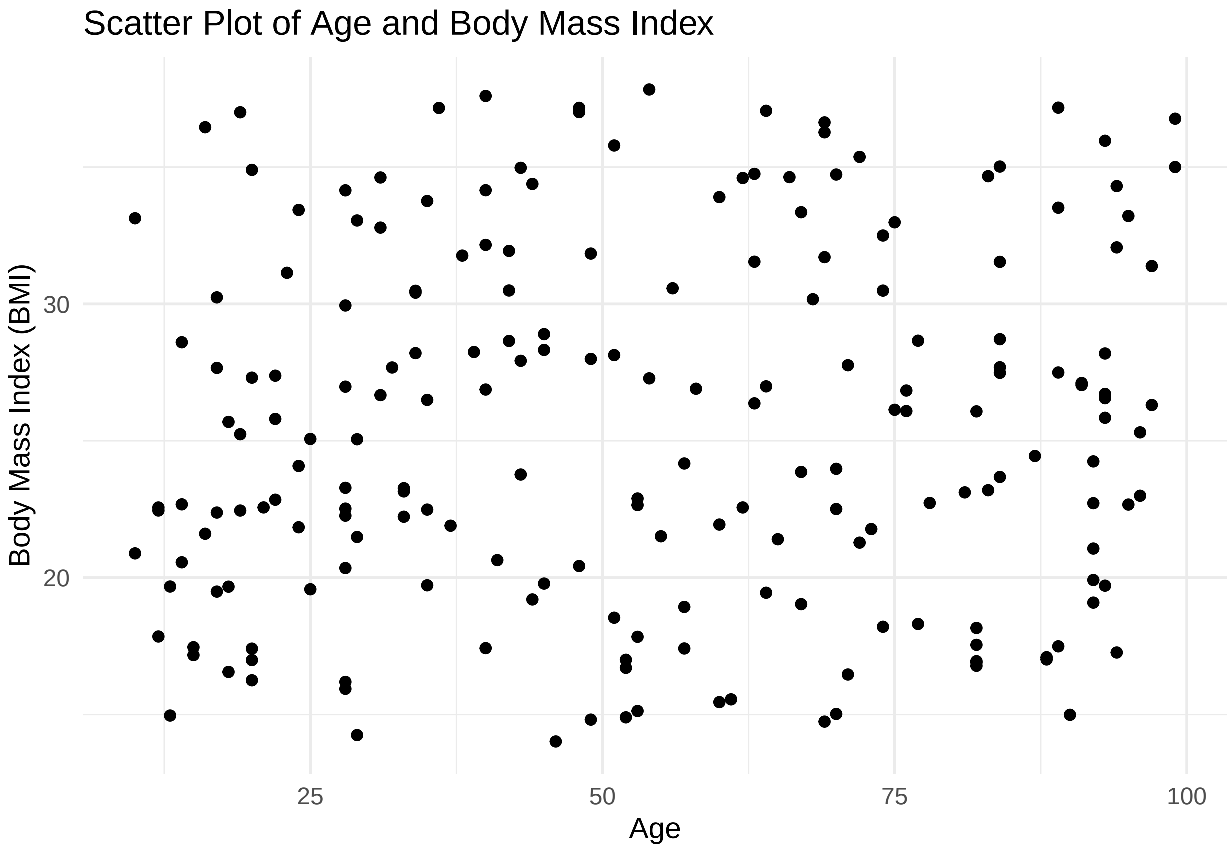
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Set random seed
set.seed(19)

# Create dataset
n <- 200

age <- sample(10:99, n, replace = TRUE)
sex <- sample(c("male", "female"), n, replace = TRUE)
education <- sample(c("no education", "primary", "secondary", "beyond secondary"), n, replace = TRUE)
socioeconomic_status <- sample(c("low", "middle", "high"), n, replace = TRUE)
bmi <- runif(n, 14, 38)

data <- data.frame(age, sex, education, socioeconomic_status, bmi)
#extracting first 10 rows using head function
head(data)
```

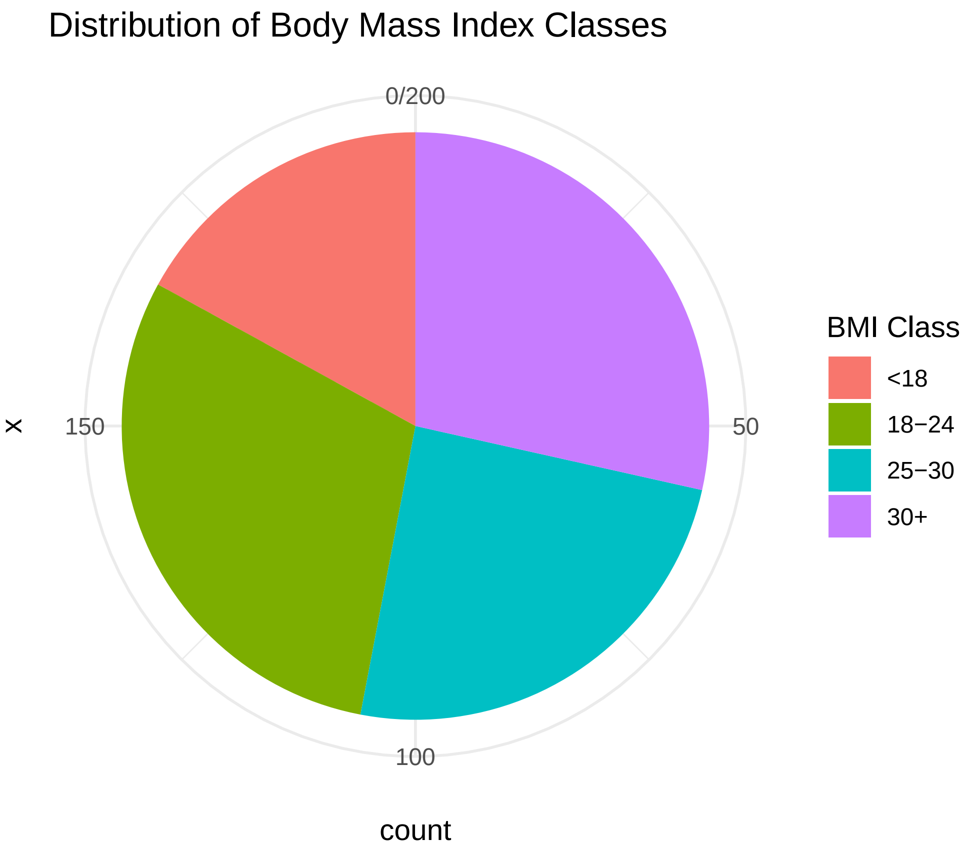
```
##   age    sex        education socioeconomic_status      bmi
## 1  63   male beyond secondary               middle 26.36727
## 2  12 female        secondary                  low 17.85263
## 3  14   male        secondary                  high 28.59932
## 4  91   male        secondary                  high 27.11116
## 5  66   male        secondary                middle 34.62911
## 6  76 female beyond secondary                   low 26.08680
```

```
#part (b)
# Scatter plot of age and BMI
scatter_plot <- ggplot(data, aes(x = age, y = bmi)) +
  geom_point() +
  labs(title = "Scatter Plot of Age and Body Mass Index",
       x = "Age",
       y = "Body Mass Index (BMI)") +
  theme_minimal()
scatter_plot
```

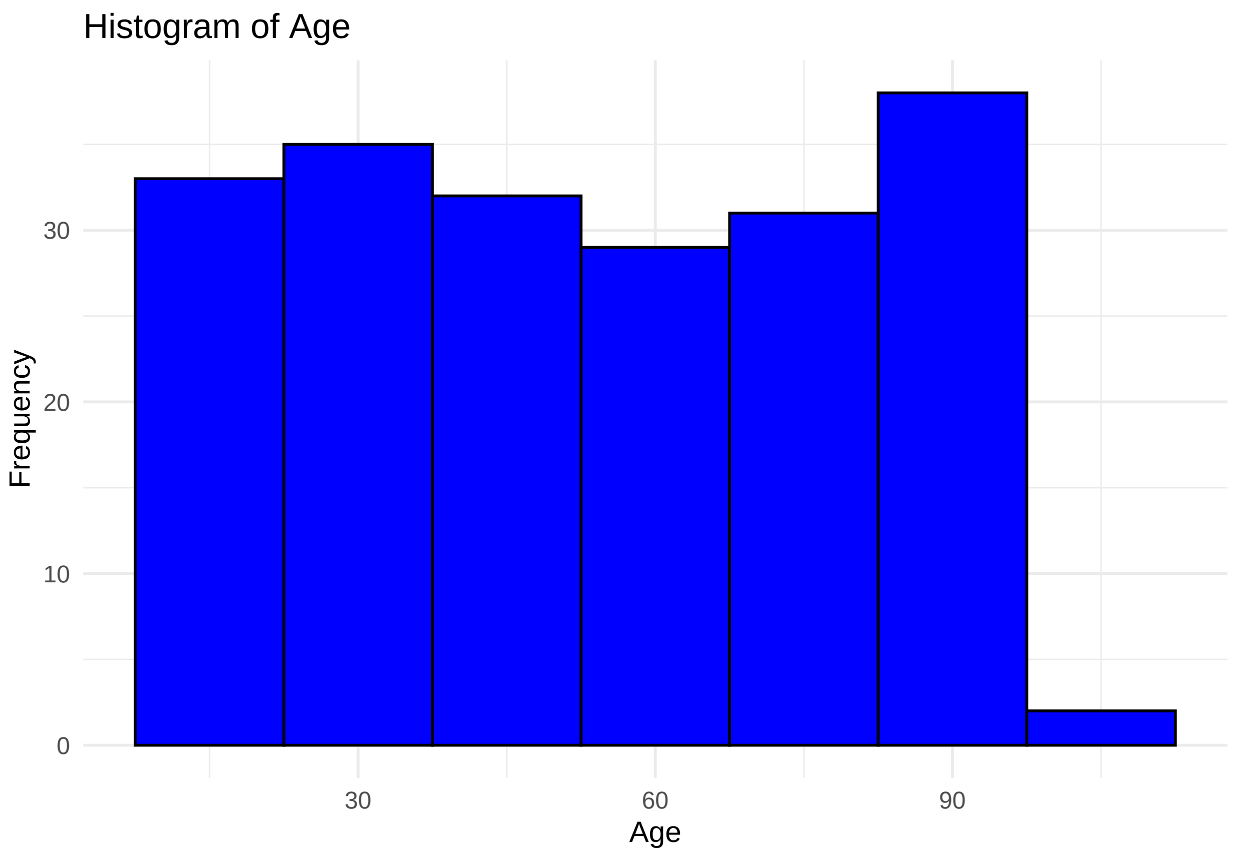


```
#The scatter plot suggests that age alone may not be a strong predictor of BMI in this dataset.


#part (c)
# Create BMI classes
data$bmi_class <- cut(data$bmi, breaks = c(-Inf, 18, 24, 30, Inf), labels = c("<18", "18-24", "25-30",

# Count the number of cases in each BMI class
bmi_class_counts <- data %>%
  group_by(bmi_class) %>%
```

```r
  summarize(count = n())

# Create pie chart
pie_chart <- ggplot(bmi_class_counts, aes(x = "", y = count, fill = bmi_class)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y") +
  labs(title = "Distribution of Body Mass Index Classes",
       fill = "BMI Class") +
  theme_minimal()

pie_chart
```



```r
#the pie chart shows that the data is well distributed among

#part(d)
# Histogram of age with bin size 15
histogram <- ggplot(data, aes(x = age)) +
  geom_histogram(binwidth = 15, fill = "blue", color = "black") +
  labs(title = "Histogram of Age",
       x = "Age",
       y = "Frequency") +
  theme_minimal()
histogram
```



```
#With a bin size of 15, there are approximately 6 bins covering the age range from 10 to 99 years.
#The histogram appears to be roughly symmetric, suggesting a relatively even distribution of
#ages in the dataset.
```

```
##Question 7
##part-a
aq<-airquality
# Perform the Shapiro-Wilk test
shapiro.test(aq$Temp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aq$Temp
## W = 0.97617, p-value = 0.009319
```

```
# If the p-value is less than the significance level (usually 0.05), we reject the null hypothesis
# and conclude that the data does not follow a normal distribution.

##part-b
# Load the necessary library
library(stats)

# Perform the Bartlett test
bartlett.test(aq$Temp ~ aq$Month)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  aq$Temp by aq$Month
## Bartlett's K-squared = 12.023, df = 4, p-value = 0.01718
```

```
# The p-value is less than the significance level (usually 0.05), we reject the null hypothesis
# and conclude that the variances are not equal.


##part-c
#Since the variances of temp are not equal by month, we can use the Welch's
#t-test, which is a variation of the t-test that does not assume equal variances.
#This test is used to compare the means of two or more groups

#pard-d
#since the variance is not equal so used kruskal test
kruskal_test <- kruskal.test(Temp ~ factor(Month), data = airquality)
kruskal_test
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Temp by factor(Month)
## Kruskal-Wallis chi-squared = 73.328, df = 4, p-value = 4.496e-15
```

```
##Question 8
#part - a
#loadning necessary packages
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
data<-Arrests
str(data)
```

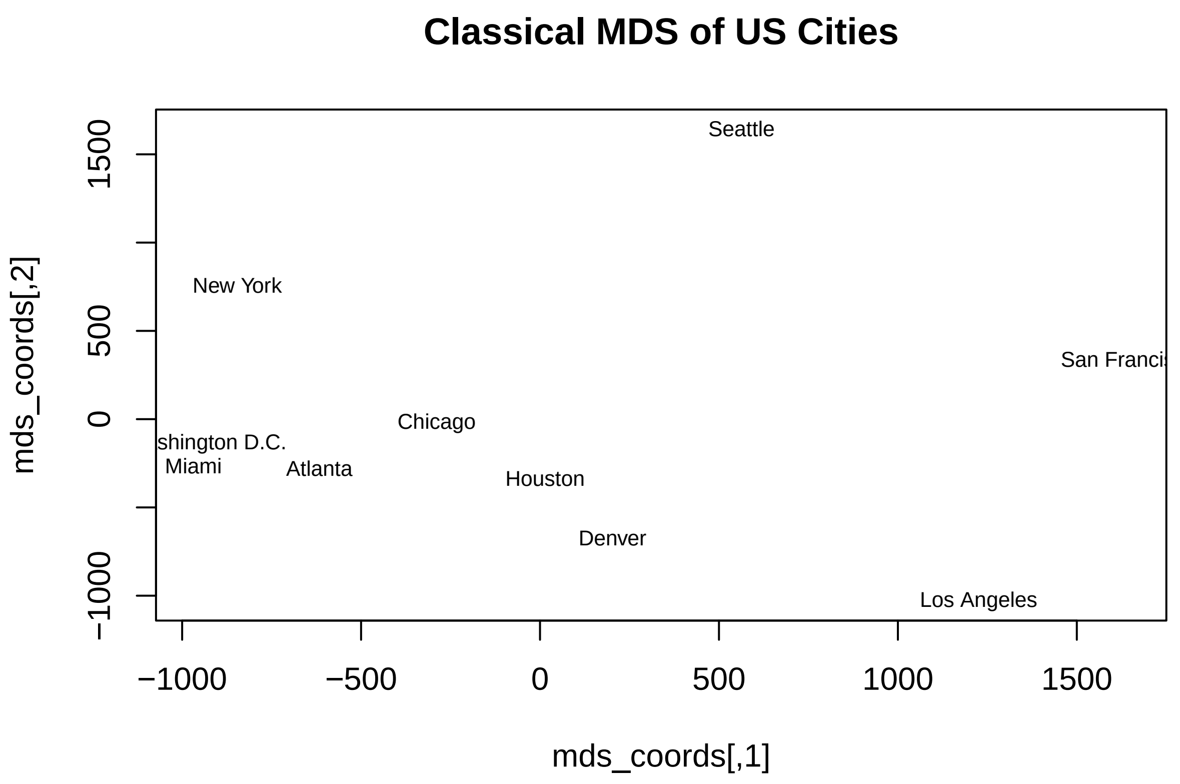
```
## 'data.frame':    5226 obs. of  8 variables:
##  $ released: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 2 2 2 ...
##  $ colour  : Factor w/ 2 levels "Black","White": 2 1 2 1 1 1 2 2 1 2 ...
##  $ year    : int  2002 1999 2000 2000 1999 1998 1999 1998 2000 2001 ...
##  $ age     : int  21 17 24 46 27 16 40 34 23 30 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 1 2 2 ...
##  $ employed: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 1 2 2 2 ...
##  $ citizen : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ checks  : int  3 3 3 1 1 0 0 1 4 3 ...
```

```r
# Split the data into training and test sets
set.seed(19)
ind<-sample(2, nrow(data), replace = T, prob = c(0.8, 0.2))
training <- data[ind==1,]
testing <- data[ind==2,]
```

```r
#Qesstion 9(OR)
library(stats)

# Question a
# Create a distance matrix from the given data in the problem
city_distances <- matrix(c(
  0, 587, 1212, 701, 1936, 604, 748, 2139, 2182, 543,
  587, 0, 920, 940, 1745, 1188, 713, 1858, 1737, 597,
  1212, 920, 0, 879, 831, 1726, 1611, 1949, 2204, 1494,
  701, 940, 879, 0, 1374, 968, 1420, 1645, 1891, 1220,
  1936, 1745, 831, 1374, 0, 2339, 2451, 347, 2734, 2300,
  604, 1188, 1726, 968, 2339, 0, 1092, 2594, 2408, 923,
  748, 713, 1611, 1420, 2451, 1092, 0, 2571, 678, 205,
  2139, 1858, 1949, 1645, 347, 2594, 2571, 0, 678, 2442,
  2182, 1737, 2204, 1891, 2734, 2408, 678, 678, 0, 2329,
  543, 597, 1494, 1220, 2300, 923, 205, 2442, 2329, 0
), nrow = 10, byrow = TRUE)
# Assigning names to row and columns
city_names <- c("Atlanta", "Chicago", "Denver", "Houston", "Los Angeles", "Miami",
                "New York", "San Francisco", "Seattle", "Washington D.C.")
rownames(city_distances) <- city_names
colnames(city_distances) <- city_names

# Convert to a dissimilarity object
city_dissimilarity <- as.dist(city_distances)

# Question b
# Fit the classical MDS model using city.dissimilarity object
mds_model <- cmdscale(city_dissimilarity, eig = TRUE, k = 2)

# Question c
# Summarizing the model
mds_coords <- mds_model$points
print(mds_coords)
```

```
##                       [,1]        [,2]
## Atlanta         -616.46326  -277.03319
## Chicago         -288.61063   -22.16151
## Denver           202.61148  -672.61019
## Houston           14.25242  -335.54496
## Los Angeles     1225.78174 -1033.78934
## Miami           -968.45797  -264.31832
## New York        -845.50822   757.66327
## San Francisco   1645.58380   339.92746
## Seattle          563.12009  1646.43854
## Washington D.C. -932.30945  -138.57175
```

```
# Question d
# Bi-plot of the model
plot(mds_coords, type = "n")
text(mds_coords, labels = city_names, cex = 0.7)
title("Classical MDS of US Cities")
```



```
#Qestion 10
# Load necessary libraries
library(datasets)
library(ggplot2)
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.3.3
```

```
library(dendextend)
```

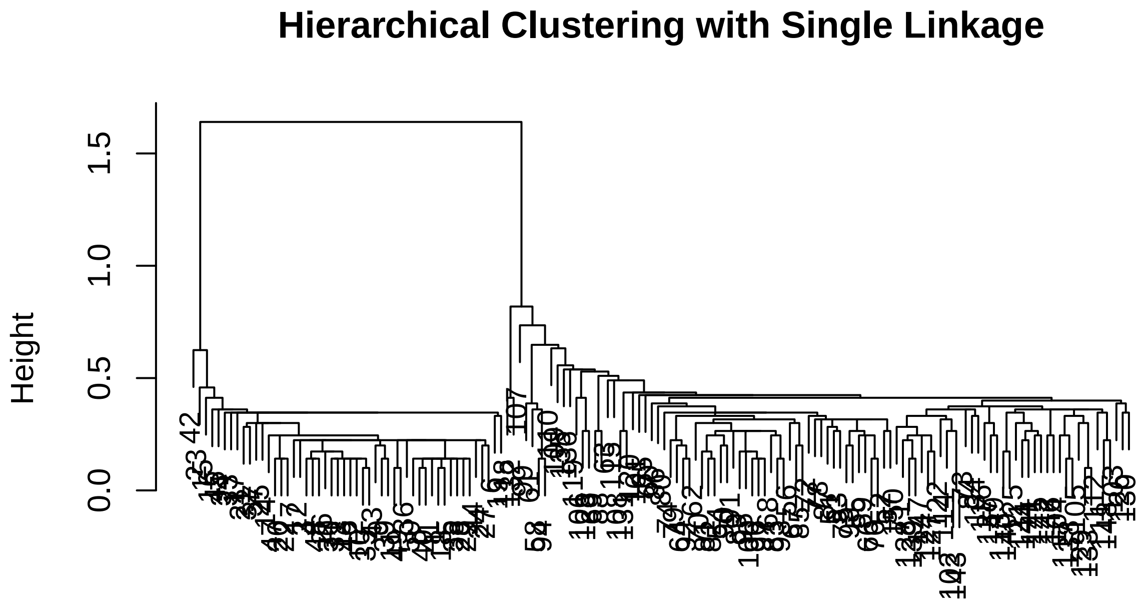
```
## Warning: package 'dendextend' was built under R version 4.3.3
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

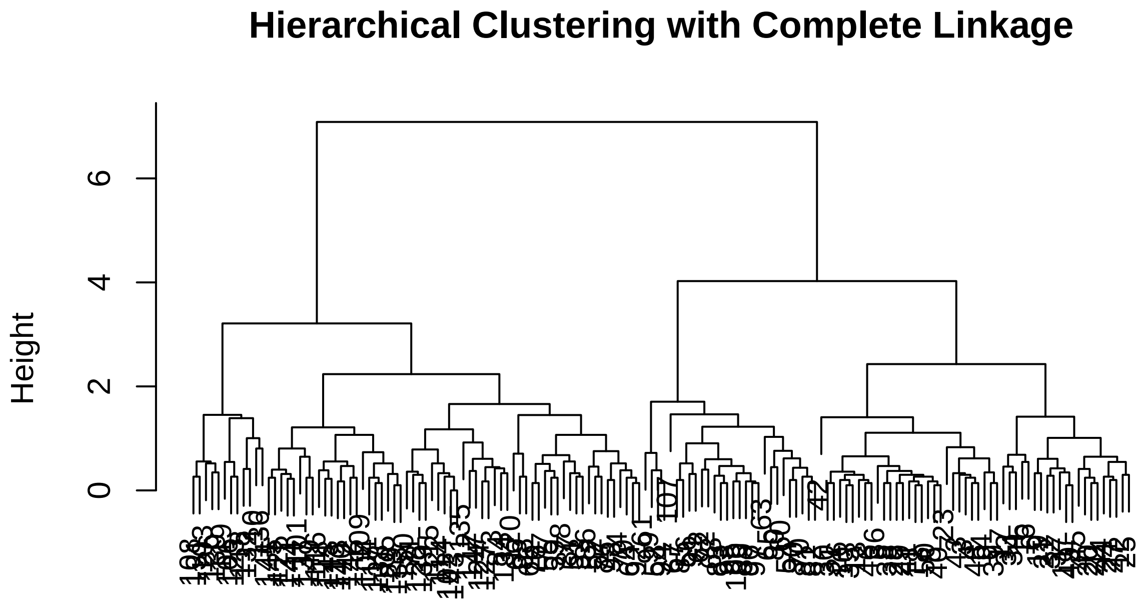
```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

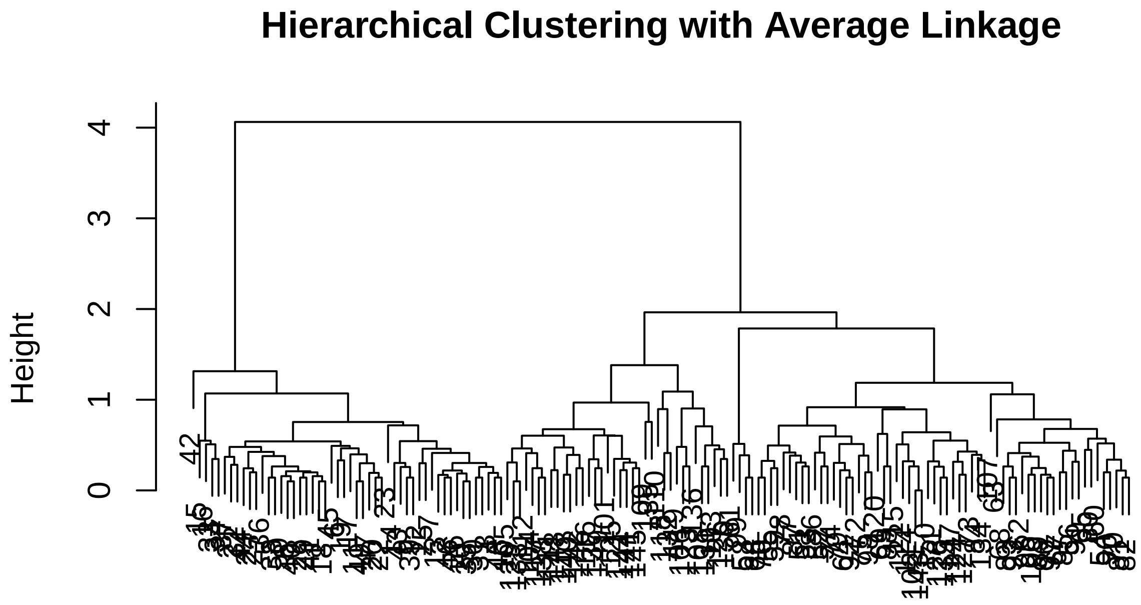
```r
# Load the iris dataset
data("iris")
iris_data <- iris[, 1:4]

# Compute the distance matrix
dist_matrix <- dist(iris_data)

# Hierarchical clustering using single linkage
hc_single <- hclust(dist_matrix, method = "single")
plot(hc_single, main = "Hierarchical Clustering with Single Linkage", xlab = "", sub = "", cex = 0.9)
```



```r
# Hierarchical clustering using complete linkage
hc_complete <- hclust(dist_matrix, method = "complete")
plot(hc_complete, main = "Hierarchical Clustering with Complete Linkage", xlab = "", sub = "", cex = 0.
```



```r
# Hierarchical clustering using average linkage
hc_average <- hclust(dist_matrix, method = "average")
plot(hc_average, main = "Hierarchical Clustering with Average Linkage", xlab = "", sub = "", cex = 0.9)
```



```r
# Calculate cophenetic correlation coefficients
coph_single <- cor(dist_matrix, cophenetic(hc_single))
coph_complete <- cor(dist_matrix, cophenetic(hc_complete))
coph_average <- cor(dist_matrix, cophenetic(hc_average))

# Print cophenetic correlation coefficients
cat("Cophenetic Correlation Coefficient for Single Linkage:", coph_single, "\n")
```

```
## Cophenetic Correlation Coefficient for Single Linkage: 0.8638787
```

```r
cat("Cophenetic Correlation Coefficient for Complete Linkage:", coph_complete, "\n")
```

```
## Cophenetic Correlation Coefficient for Complete Linkage: 0.7269857
```

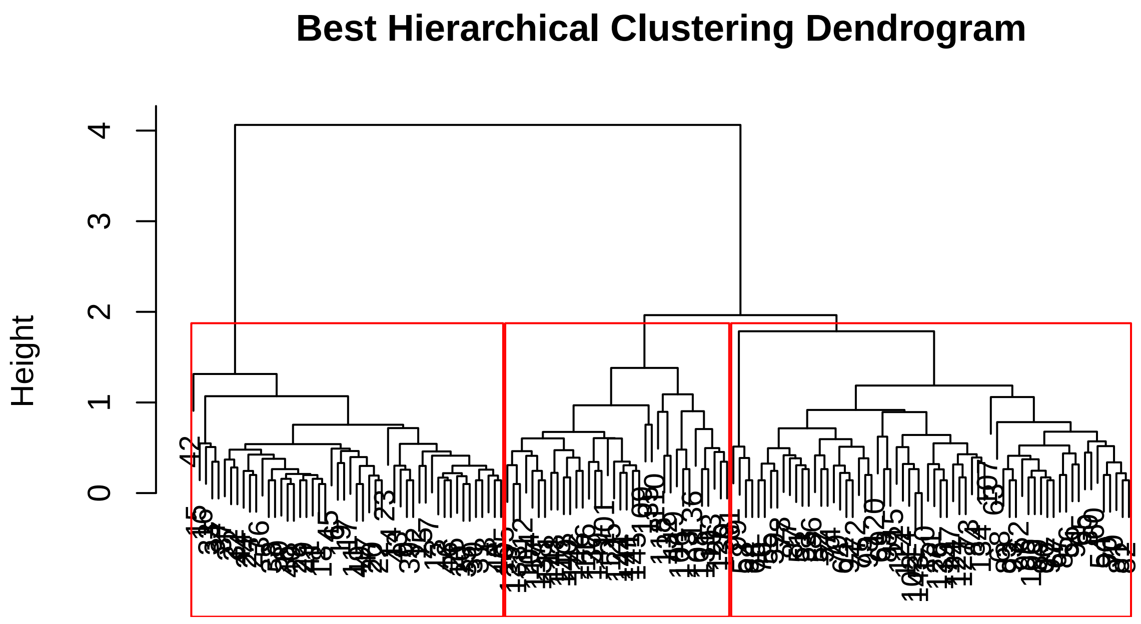
```r
cat("Cophenetic Correlation Coefficient for Average Linkage:", coph_average, "\n")
```

```
## Cophenetic Correlation Coefficient for Average Linkage: 0.8769561
```

```r
# Determine the best model
best_model <- which.max(c(coph_single, coph_complete, coph_average))
if (best_model == 1) {
  cat("Best model: Single Linkage\n")
  best_hc <- hc_single
} else if (best_model == 2) {
  cat("Best model: Complete Linkage\n")
  best_hc <- hc_complete
} else {
  cat("Best model: Average Linkage\n")
  best_hc <- hc_average
}
```

```
## Best model: Average Linkage
```

```
# Plot the best dendrogram
plot(best_hc, main = "Best Hierarchical Clustering Dendrogram", xlab = "", sub = "", cex = 0.9)

# Determine the number of clusters
rect.hclust(best_hc, k = 3, border = "red")
```



```
clusters <- cutree(best_hc, k = 3)
table(clusters)
```

```
## clusters
##  1  2  3
## 50 64 36
```