

LAB 6

GYANADA

21BCE7727

1. Cluster a mushroom dataset using k-means algorithm and visualize using Quadtree.

```
In [119...]: data <- read.csv("C:/Users/gyanada/Downloads/mushroom/mushrooms.csv")
```

```
In [120...]: head(data)
```

A data.frame: 6 × 12

	class	cap.shape	cap.surface	cap.color	bruises	odor	gill.attachment	gill.spacing	gill.size	gill.color	stalk.surface.below.ring	stalk.color
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	p	x	s	n	t	p	f	c	n	k	...	s
2	e	x	s	y	t	a	f	c	b	k	...	s
3	e	b	s	w	t	l	f	c	b	n	...	s
4	p	x	y	w	t	p	f	c	n	n	...	s
5	e	x	s	g	f	n	f	w	b	k	...	s
6	e	x	y	y	t	a	f	c	b	n	...	s

```
In [121...]: summary(data)
```

```
  class          cap.shape        cap.surface      cap.color
Length:8124    Length:8124    Length:8124    Length:8124
Class :character Class :character Class :character Class :character
Mode :character  Mode :character  Mode :character  Mode :character
  bruises        odor          gill.attachment  gill.spacing
Length:8124    Length:8124    Length:8124    Length:8124
Class :character Class :character Class :character Class :character
Mode :character  Mode :character  Mode :character  Mode :character
  gill.size      gill.color     stalk.shape     stalk.root
Length:8124    Length:8124    Length:8124    Length:8124
Class :character Class :character Class :character Class :character
Mode :character  Mode :character  Mode :character  Mode :character
  stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring
Length:8124          Length:8124          Length:8124
Class :character          Class :character          Class :character
Mode :character          Mode :character          Mode :character
  stalk.color.below.ring  veil.type       veil.color
Length:8124          Length:8124          Length:8124
Class :character          Class :character          Class :character
Mode :character          Mode :character          Mode :character
  ring.number       ring.type       spore.print.color  population
Length:8124          Length:8124          Length:8124    Length:8124
Class :character          Class :character          Class :character
Mode :character          Mode :character          Mode :character
  habitat
```

```
Length:8124
Class :character
Mode :character
```

```
In [122...]: dim(data)
```

```
8124 · 23
```

```
In [123...]: x <- is.na(data)
```

```
In [124...]: data <- na.omit(data)
```

```
In [125...]: str(data)
```

```
'data.frame': 8124 obs. of 23 variables:  
 $ class           : chr  "p" "e" "e" "p" ...  
 $ cap.shape       : chr  "x" "x" "b" "x" ...  
 $ cap.surface     : chr  "s" "s" "s" "y" ...  
 $ cap.color       : chr  "n" "y" "w" "w" ...  
 $ bruises         : chr  "t" "t" "t" "t" ...  
 $ odor            : chr  "p" "a" "l" "p" ...  
 $ gill.attachment: chr  "f" "f" "f" "f" ...  
 $ gill.spacing    : chr  "c" "c" "c" "c" ...  
 $ gill.size       : chr  "n" "b" "b" "n" ...  
 $ gill.color      : chr  "k" "k" "n" "n" ...  
 $ stalk.shape     : chr  "e" "e" "e" "e" ...  
 $ stalk.root      : chr  "e" "c" "c" "e" ...  
 $ stalk.surface.above.ring: chr  "s" "s" "s" "s" ...  
 $ stalk.surface.below.ring: chr  "s" "s" "s" "s" ...  
 $ stalk.color.above.ring : chr  "w" "w" "w" "w" ...  
 $ stalk.color.below.ring : chr  "w" "w" "w" "w" ...  
 $ veil.type       : chr  "p" "p" "p" "p" ...  
 $ veil.color      : chr  "w" "w" "w" "w" ...  
 $ ring.number     : chr  "o" "o" "o" "o" ...  
 $ ring.type       : chr  "p" "p" "p" "p" ...  
 $ spore.print.color: chr  "k" "n" "n" "k" ...  
 $ population      : chr  "s" "n" "n" "s" ...  
 $ habitat          : chr  "u" "g" "m" "u" ...
```

In [126...]

```
print(colnames(data))
```

```
[1] "class"           "cap.shape"  
[3] "cap.surface"    "cap.color"  
[5] "bruises"         "odor"  
[7] "gill.attachment" "gill.spacing"  
[9] "gill.size"       "gill.color"  
[11] "stalk.shape"     "stalk.root"  
[13] "stalk.surface.above.ring" "stalk.surface.below.ring"  
[15] "stalk.color.above.ring" "stalk.color.below.ring"  
[17] "veil.type"       "veil.color"  
[19] "ring.number"     "ring.type"  
[21] "spore.print.color" "population"  
[23] "habitat"
```

```
In [127...]  
if (is.data.frame(data) && 'class' %in% names(data)) {  
    unique_classes <- unique(data$class)  
    print(unique_classes)  
} else {  
    print("Error: 'data' is not a data frame or does not have a 'class' column.")  
}  
  
[1] "p" "e"
```

```
In [128...]  
library(dplyr)  
for (col in names(data)) {  
    if (length(unique(data[[col]])) == 2) {  
        data[[col]] <- as.factor(data[[col]])  
        data[[col]] <- as.numeric(data[[col]]) - 1  
    }  
}
```

```
In [129...]  
constant_vars <- sapply(data, function(x) length(unique(x)) == 1)  
constant_vars_names <- names(data)[constant_vars]  
data <- data[, !constant_vars]  
dummy_data <- model.matrix(~ . - 1, data = data)  
dummy_data <- as.data.frame(dummy_data)  
head(dummy_data)
```

A data.frame: 6 × 97

	class	cap.shapeb	cap.shapec	cap.shapef	cap.shapek	cap.shapes	cap.shapex	cap.surfaceg	cap.surfaces	cap.surfacey	...	populatio
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	...	<dbl>
1	1	0	0	0	0	0	1	0	1	0	0	...
2	0	0	0	0	0	0	1	0	1	0	0	...
3	0	1	0	0	0	0	0	0	1	0	0	...
4	1	0	0	0	0	0	1	0	0	0	1	...
5	0	0	0	0	0	0	1	0	1	0	0	...
6	0	0	0	0	0	0	1	0	0	1	0	...

```
In [130...]  
print(class(data))  
head(data)
```

```
[1] "data.frame"
```

A data.fra

	class	cap.shape	cap.surface	cap.color	bruises	odor	gill.attachment	gill.spacing	gill.size	gill.color	...	stalk.surface.above.ring
	<dbl>	<chr>	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<chr>	...	<chr>
1	1	x	s	n	1	p	1	0	1	k	...	s
2	0	x	s	y	1	a	1	0	0	k	...	s
3	0	b	s	w	1	l	1	0	0	n	...	s
4	1	x	y	w	1	p	1	0	1	n	...	s
5	0	x	s	g	0	n	1	1	0	k	...	s
6	0	x	y	y	1	a	1	0	0	n	...	s

```
In [131...]: X <- data[, !names(df) %in% "class"]  
head(X)
```

A
data.frame
6 × 0

1
2
3
4
5
6

```
In [132]: Y <- data$class`  
head(Y)
```

1:0:0:1:0:0

```
In [133]: Y <- data.frame(Y)  
head(Y)
```

A
data.frame:
6 × 1

	Y
<dbl>	
1	1
2	0
3	0
4	1
5	0
6	0

In [134...]

```
summary(data)
```

class	cap.shape	cap.surface	cap.color
Min. :0.000	Length:8124	Length:8124	Length:8124
1st Qu.:0.000	Class :character	Class :character	Class :character
Median :0.000	Mode :character	Mode :character	Mode :character
Mean :0.482			
3rd Qu.:1.000			
Max. :1.000			
bruises	odor	gill.attachment	gill.spacing
Min. :0.0000	Length:8124	Min. :0.0000	Min. :0.0000
1st Qu.:0.0000	Class :character	1st Qu.:1.0000	1st Qu.:0.0000
Median :0.0000	Mode :character	Median :1.0000	Median :0.0000
Mean :0.4156		Mean :0.9742	Mean :0.1615
3rd Qu.:1.0000		3rd Qu.:1.0000	3rd Qu.:0.0000
Max. :1.0000		Max. :1.0000	Max. :1.0000
gill.size	gill.color	stalk.shape	stalk.root
Min. :0.0000	Length:8124	Min. :0.0000	Length:8124
1st Qu.:0.0000	Class :character	1st Qu.:0.0000	Class :character
Median :0.0000	Mode :character	Median :1.0000	Mode :character
Mean :0.3092		Mean :0.5672	
3rd Qu.:1.0000		3rd Qu.:1.0000	
Max. :1.0000		Max. :1.0000	
stalk.surface.above.ring	stalk.surface.below.ring	stalk.color.above.ring	
Length:8124	Length:8124	Length:8124	
Class :character	Class :character	Class :character	
Mode :character	Mode :character	Mode :character	

stalk.color.below.ring	veil.color	ring.number	
Length:8124	Length:8124	Length:8124	
Class :character	Class :character	Class :character	
Mode :character	Mode :character	Mode :character	

ring.type	spore.print.color	population	habitat
Length:8124	Length:8124	Length:8124	Length:8124
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

```
In [135...]: install.packages("cluster")
library(cluster)
```

Warning message:
"package 'cluster' is in use and will not be installed"

```
In [136...]: any(is.na(data))
```

FALSE

```
In [137...]: nan_values <- sapply(data, function(x) any(is.nan(x)))
any(nan_values)
```

FALSE

```
In [138...]: infinite_values <- sapply(data, function(x) any(is.infinite(x)))
any(infinite_values)
```

FALSE

```
In [139...]: # Identify character columns
char_columns <- names(data)[sapply(data, is.character)]

# One-hot encode character columns
encoded_data <- data
for (col in char_columns) {
  encoded_data <- cbind(encoded_data, model.matrix(~ data[[col]] - 1))
}
encoded_data <- encoded_data[, !names(encoded_data) %in% char_columns]

# Convert data frame to numeric
encoded_data <- as.data.frame(sapply(encoded_data, as.numeric))

# Check the structure of the encoded data
str(encoded_data)
```

```
'data.frame': 8124 obs. of 112 variables:  
$ class : num 1 0 0 1 0 0 0 0 1 0 ...  
$ bruises : num 1 1 1 1 0 1 1 1 1 1 ...  
$ gill.attachment: num 1 1 1 1 1 1 1 1 1 1 ...  
$ gill.spacing : num 0 0 0 0 1 0 0 0 0 0 ...  
$ gill.size : num 1 0 0 1 0 0 0 0 1 0 ...  
$ stalk.shape : num 0 0 0 0 1 0 0 0 0 0 ...  
$ data[[col]]b : num 0 0 1 0 0 0 1 1 0 1 ...  
$ data[[col]]c : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]f : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]k : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]s : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]x : num 1 1 0 1 1 1 0 0 1 0 ...  
$ data[[col]]f.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]g : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]s.1 : num 1 1 1 0 1 0 1 0 0 1 ...  
$ data[[col]]y : num 0 0 0 1 0 1 0 1 1 0 ...  
$ data[[col]]b.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]c.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]e : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]g.1 : num 0 0 0 0 1 0 0 0 0 0 ...  
$ data[[col]]n : num 1 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]p : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]r : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]u : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]w : num 0 0 1 1 0 0 1 1 1 0 ...  
$ data[[col]]y.1 : num 0 1 0 0 0 1 0 0 0 1 ...  
$ data[[col]]a : num 0 1 0 0 0 1 1 0 0 1 ...  
$ data[[col]]c.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]f.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]l : num 0 0 1 0 0 0 0 1 0 0 ...  
$ data[[col]]m : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]n.1 : num 0 0 0 0 1 0 0 0 0 0 ...  
$ data[[col]]p.1 : num 1 0 0 1 0 0 0 0 1 0 ...  
$ data[[col]]s.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]y.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]b.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]e.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]g.2 : num 0 0 0 0 0 0 1 0 0 1 ...  
$ data[[col]]h : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]k.1 : num 1 1 0 0 1 0 0 0 0 0 ...  
$ data[[col]]n.2 : num 0 0 1 1 0 1 0 1 0 0 ...  
$ data[[col]]o : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]p.2 : num 0 0 0 0 0 0 0 0 1 0 ...
```

```
$ data[[col]]r.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]u.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]w.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]y.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]? : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]b.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]c.3 : num 0 1 1 0 0 1 1 1 0 1 ...  
$ data[[col]]e.2 : num 1 0 0 1 1 0 0 0 1 0 ...  
$ data[[col]]r.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]f.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]k.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]s.3 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]y.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]f.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]k.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]s.4 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]y.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]b.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]c.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]e.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]g.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]n.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]o.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]p.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]w.2 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]y.6 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]b.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]c.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]e.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]g.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]n.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]o.2 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]p.4 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]w.3 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]y.7 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]n.5 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]o.3 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]w.4 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]y.8 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]n.6 : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]o.4 : num 1 1 1 1 1 1 1 1 1 1 ...  
$ data[[col]]t : num 0 0 0 0 0 0 0 0 0 0 ...  
$ data[[col]]e.5 : num 0 0 0 1 0 0 0 0 0 0 ...  
$ data[[col]]f.5 : num 0 0 0 0 0 0 0 0 0 0 ...
```

```
$ data[[col]]l.1 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]n.7 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]p.5 : num  1 1 1 1 0 1 1 1 1 1 ...
$ data[[col]]b.6 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]h.1 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]k.4 : num  1 0 0 1 0 1 1 0 1 1 ...
$ data[[col]]n.8 : num  0 1 1 0 1 0 0 1 0 0 ...
$ data[[col]]o.5 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]r.3 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]u.2 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]w.5 : num  0 0 0 0 0 0 0 0 0 0 ...
$ data[[col]]y.9 : num  0 0 0 0 0 0 0 0 0 0 ...
[list output truncated]
```

In [140...]

```
# Check data types of each column
column_types <- sapply(data, class)

# Print data types of each column
print(column_types)

# Convert non-numeric columns to numeric
for (col in names(data)) {
  if (!is.numeric(data[[col]])) {
    data[[col]] <- as.numeric(data[[col]])
  }
}

# Check data types again
column_types_updated <- sapply(data, class)
print(column_types_updated)
```

class	cap.shape	cap.surface
"numeric"	"character"	"character"
cap.color	bruises	odor
"character"	"numeric"	"character"
gill.attachment	gill.spacing	gill.size
"numeric"	"numeric"	"numeric"
gill.color	stalk.shape	stalk.root
"character"	"numeric"	"character"
stalk.surface.above.ring	stalk.surface.below.ring	stalk.color.above.ring
"character"	"character"	"character"
stalk.color.below.ring	veil.color	ring.number
"character"	"character"	"character"
ring.type	spore.print.color	population
"character"	"character"	"character"
habitat		
"character"		

```
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"  
Warning message in eval(expr, envir, enclos):  
"NAs introduced by coercion"
```

```
      class          cap.shape        cap.surface
      "numeric"      "numeric"
      cap.color      bruises           odor
      "numeric"      "numeric"
gill.attachment   gill.spacing     gill.size
      "numeric"      "numeric"
      gill.color      stalk.shape    stalk.root
      "numeric"      "numeric"
stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring
      "numeric"      "numeric"       "numeric"
      stalk.color.below.ring
      "numeric"      veil.color    ring.number
      "numeric"      "numeric"       "numeric"
      ring.type      spore.print.color population
      "numeric"      "numeric"       "numeric"
      habitat
      "numeric"
```

```
In [141... data<-na.omit(encoded_data)
```

```
In [142... nan_values <- sapply(encoded_data, function(x) any(is.nan(x)))
any(nan_values)
```

FALSE

```
In [143... infinite_values <- sapply(encoded_data, function(x) any(is.infinite(x)))
any(infinite_values)
```

FALSE

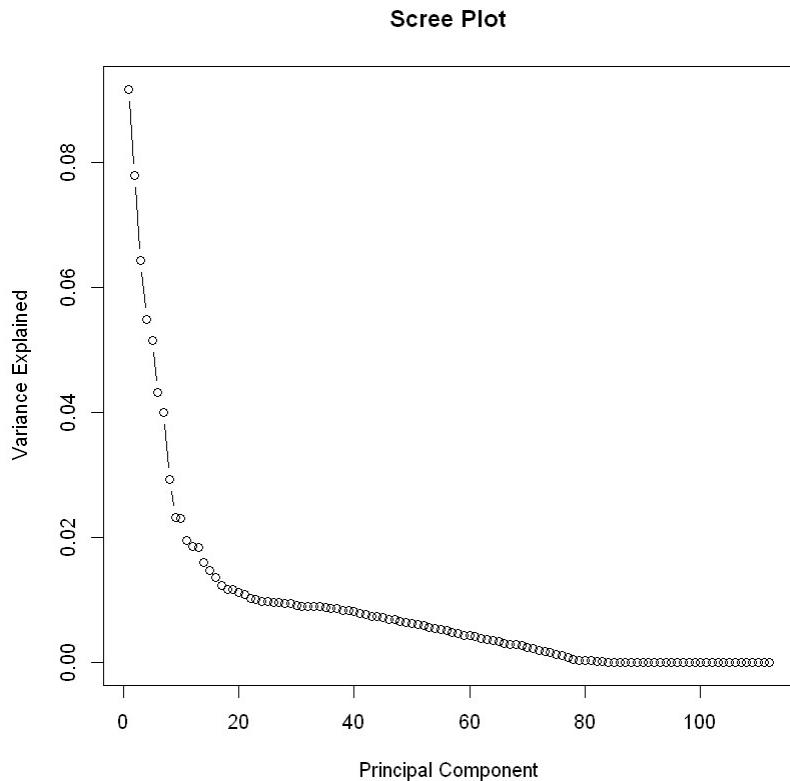
In [144...]

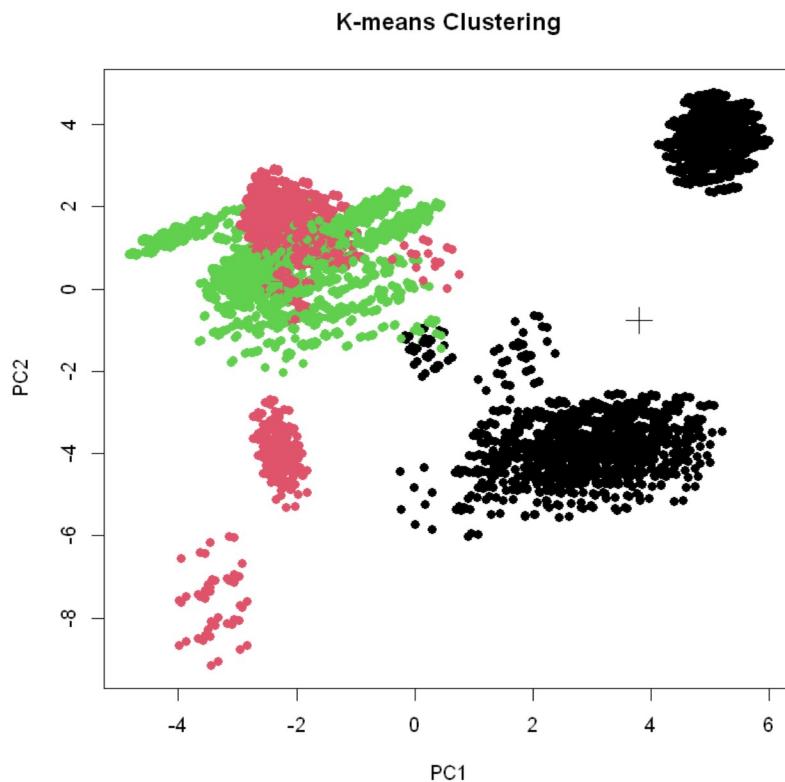
```
library(stats)
library(cluster)

pca_result <- prcomp(encoded_data, scale. = TRUE)

variance_explained <- summary(pca_result)$importance[2,]

plot(variance_explained, type = "b", main = "Scree Plot", xlab = "Principal Component", ylab = "Variance Explained")
cumulative_variance <- cumsum(variance_explained)
num_components <- which.max(cumulative_variance > 0.95)
pca_data <- as.data.frame(predict(pca_result, newdata = encoded_data)[,1:num_components])
kmeans_result <- kmeans(pca_data, centers = 3)
pca_data$cluster <- as.factor(kmeans_result$cluster)
if (num_components >= 2) {
    plot(pca_data[,1:2], col = pca_data$cluster, pch = 19, main = "K-means Clustering", xlab = "PC1", ylab = "PC2")
    points(kmeans_result$centers[,1:2], col = 1:3, pch = 3, cex = 2)
} else {
    cat("Number of principal components is less than 2. Unable to visualize clustering result.")
}
```





```
In [145...]:  
install.packages("quadtree")  
library(quadtree)
```

Warning message:
"package 'quadtree' is in use and will not be installed"

```
In [146...]:  
create_quadtree <- function(centroids, cluster_sizes, split_threshold = 1) {  
  points <- cbind(centroids, size = cluster_sizes)  
  qt <- quadtree(points, split_threshold = split_threshold)  
  return(qt)  
}
```

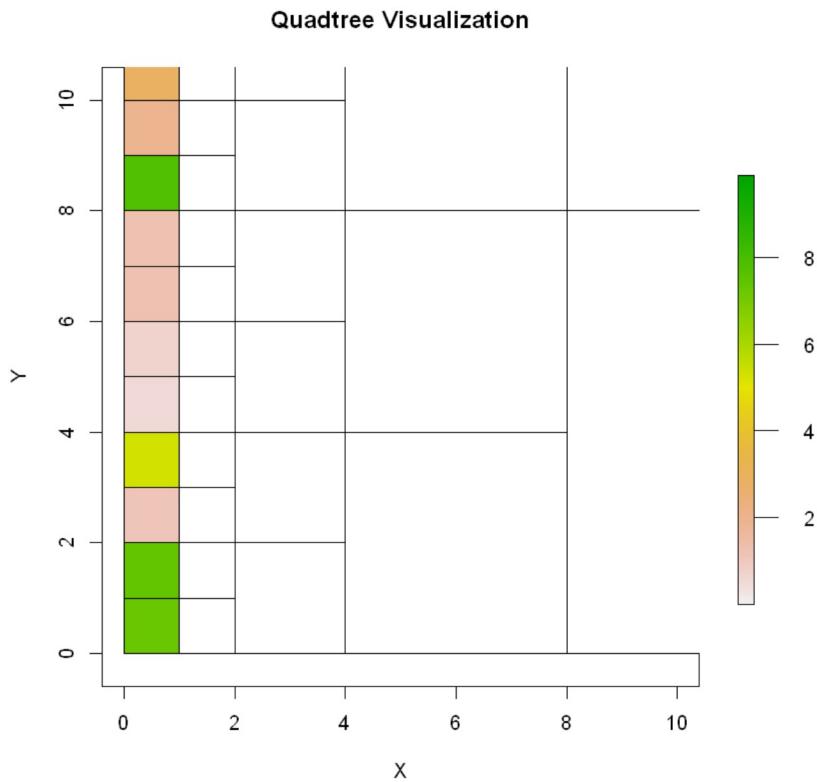
```
In [147...]:  
set.seed(42)  
encoded_data <- data.frame(x = rnorm(100), y = rnorm(100))
```

```
In [148... kmeans_result <- kmeans(encoded_data, centers = 3)
```

```
In [149... centroids <- kmeans_result$centers  
cluster_sizes <- table(kmeans_result$cluster)
```

```
In [150... xmin <- min(encoded_data$x)  
xmax <- max(encoded_data$x)  
ymin <- min(encoded_data$y)  
ymax <- max(encoded_data$y)
```

```
In [151... set.seed(42)  
encoded_data <- data.frame(x = runif(100 * 2, 0, 10))  
encoded_matrix <- as.matrix(encoded_data)  
tree <- quadtree(encoded_matrix, split_threshold = 10)  
plot(tree, main = "Quadtree Visualization", xlab = "X", ylab = "Y",  
     xlim = c(0, 10), ylim = c(0, 10), cex = 2, pch = 16)
```



```
qt <- create_quadtree(centroids, cluster_sizes) plot(qt, main = "Quadtree Visualization of Clusters", col = as.factor(kmeans_result$cluster))
```

In [152]:

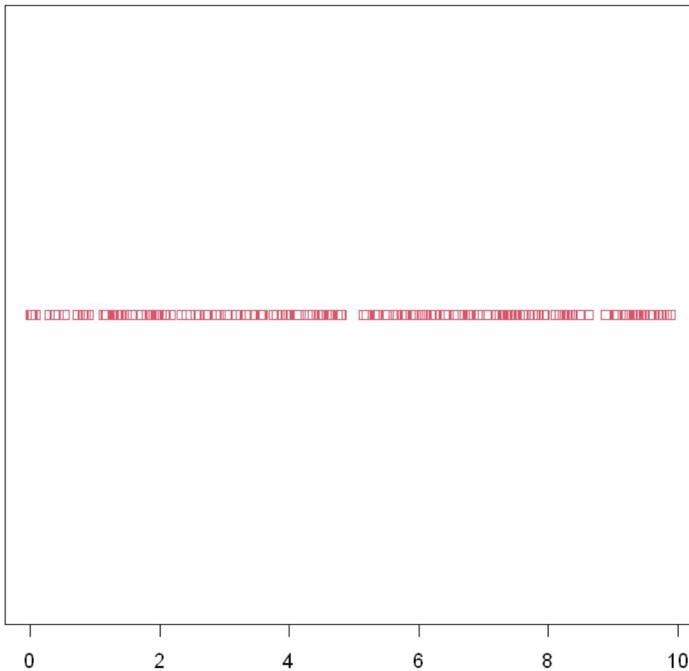
```
library(cluster)

data <- encoded_data
kmeans_result <- kmeans(data, centers = 3)

plot(data, col = kmeans_result$cluster, main = "K-means Clustering of Mushroom Data")

points(kmeans_result$centers, col = 1:25, pch = 8, cex = 2)
```

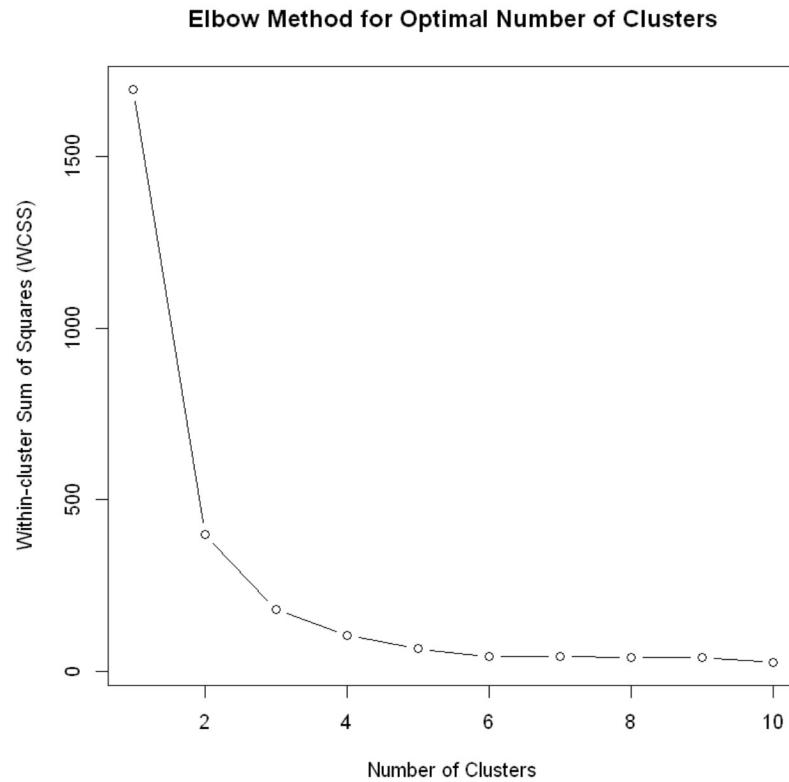
K-means Clustering of Mushroom Data



In [153...]

```
library(cluster)
data <-encoded_data
k_max <- 10
wcss <- numeric(k_max)
for (k in 1:k_max) {
  kmeans_result <- kmeans(data, centers = k)
  wcss[k] <- sum(kmeans_result$withinss)
}

plot(1:k_max, wcss, type = "b", xlab = "Number of Clusters", ylab = "Within-cluster Sum of Squares (WCSS)",
  main = "Elbow Method for Optimal Number of Clusters")
```



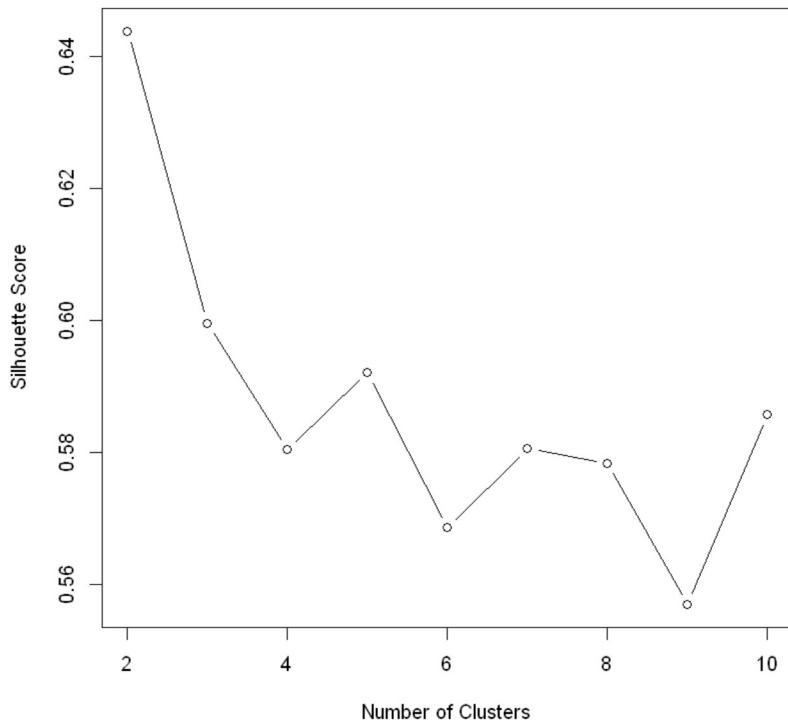
```
In [154...]: library(cluster)
library(fpc)

data <- encoded_data
k_max <- 10

silhouette_scores <- numeric(k_max)

for (k in 2:k_max) {
  kmeans_result <- kmeans(data, centers = k)
  silhouette_scores[k] <- cluster.stats(dist(data), kmeans_result$cluster)$avg.silwidth
}

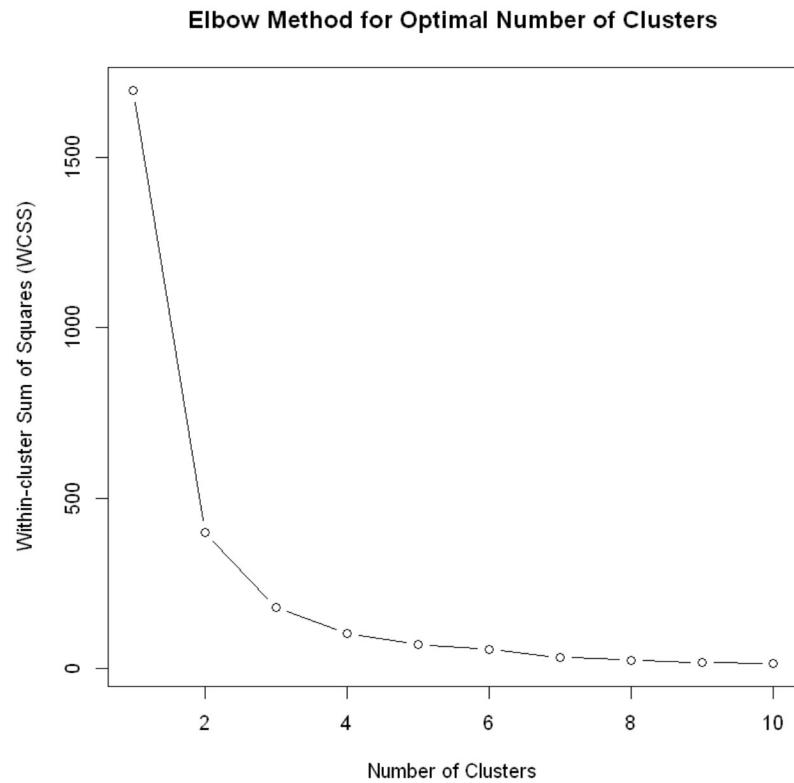
plot(2:k_max, silhouette_scores[2:k_max], type = "b",
     xlab = "Number of Clusters", ylab = "Silhouette Score",
     main = "Silhouette Method for Optimal Number of Clusters")
```

Silhouette Method for Optimal Number of Clusters

In [155...]

```
library(cluster)
data <- encoded_data
k_max <- 10
wcss <- numeric(k_max)
for (k in 1:k_max) {
  kmeans_result <- kmeans(data, centers = k)
  wcss[k] <- sum(kmeans_result$withinss)
}

plot(1:k_max, wcss, type = "b", xlab = "Number of Clusters", ylab = "Within-cluster Sum of Squares (WCSS)",
     main = "Elbow Method for Optimal Number of Clusters")
```



2. Cluster IRIS dataset using k-means algorithm and visualize using Quadtree.

In [156...]: `data <- iris`

In [157...]: `head(data)`

A data.frame: 6 × 5

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

In [158... `dim(data)`

150 · 5

In [159... `summary(data)`

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300
Median :5.800	Median :3.000	Median :4.350	Median :1.300
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500

Species
setosa :50
versicolor:50
virginica :50

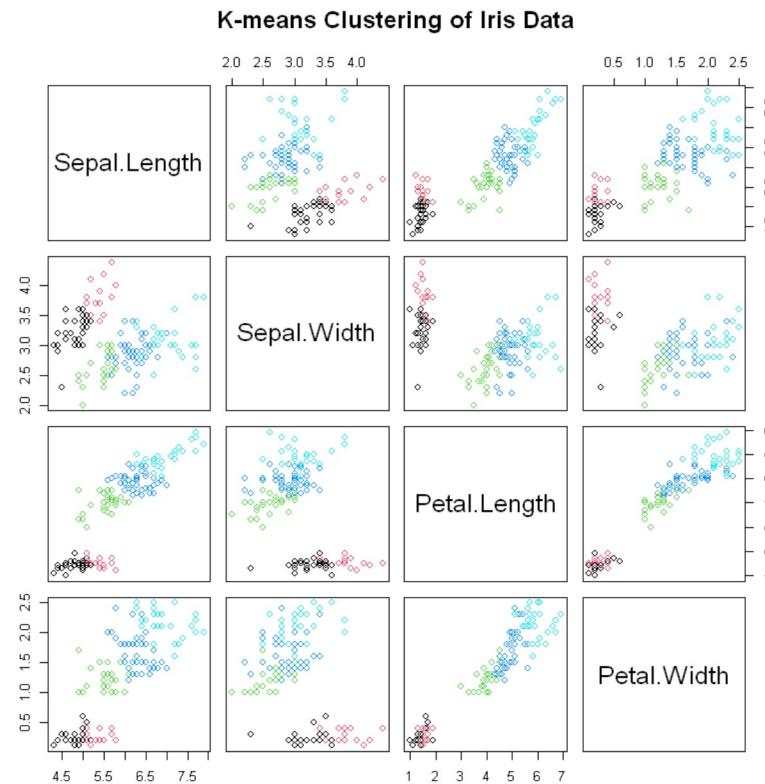
In [160... `data <- na.omit(data)`In [161... `str(data)`

```
'data.frame': 150 obs. of 5 variables:  
$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
$ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
In [162]: data <- iris[, 1:4]
```

```
In [163]: kmeans_result <- kmeans(data, centers = 5)  
centroids <- kmeans_result$centers  
cluster_sizes <- table(kmeans_result$cluster)
```

```
In [164]: plot(data, col = kmeans_result$cluster, main = "K-means Clustering of Iris Data")  
points(centroids, col = 1:4, pch = 8, cex = 2)
```



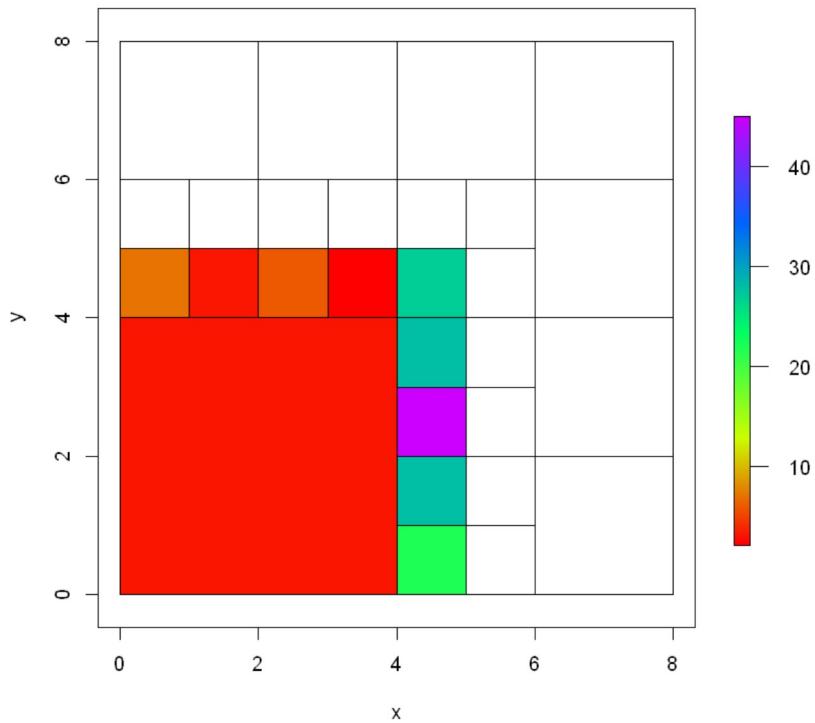
```
In [165...]  
library(cluster)  
library(quadtree)
```

```
In [166...]  
centroids <- kmeans_result$centers  
cluster_sizes <- table(kmeans_result$cluster)
```

```
In [167...]  
create_quadtree <- function(centroids, cluster_sizes, split_threshold = 1) {  
  points <- cbind(centroids, size = cluster_sizes)  
  qt <- quadtree(points, split_threshold = split_threshold)  
  return(qt)  
}
```

```
In [168...]  
data("iris")  
k <- 5  
kmeans_result <- kmeans(iris[, -5], centers = k)  
centroids <- kmeans_result$centers  
cluster_sizes <- table(kmeans_result$cluster)  
qt <- create_quadtree(centroids, cluster_sizes, split_threshold = 10)  
cluster_colors <- rainbow(k)  
plot(qt, main = "Quadtree Visualization of K-means Clustering of Iris Data", col = cluster_colors)
```

Quadtree Visualization of K-means Clustering of Iris Data



In [169...]

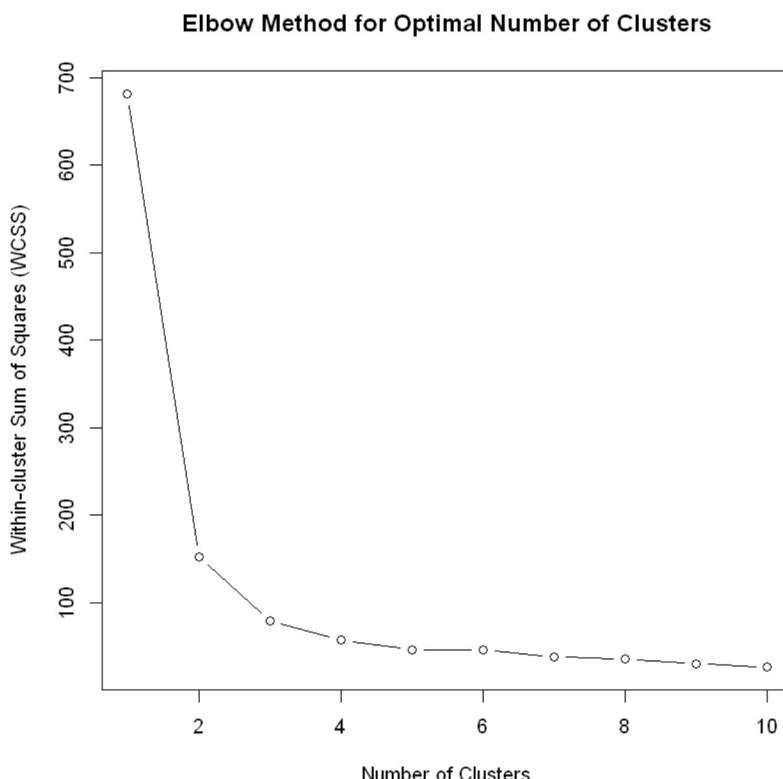
```
library(cluster)
install.packages("factoextra")
library(factoextra)
install.packages("fpc")
library(fpc)
```

Warning message:
"package 'factoextra' is in use and will not be installed"
Warning message:
"package 'fpc' is in use and will not be installed"

In []:

In [170...]

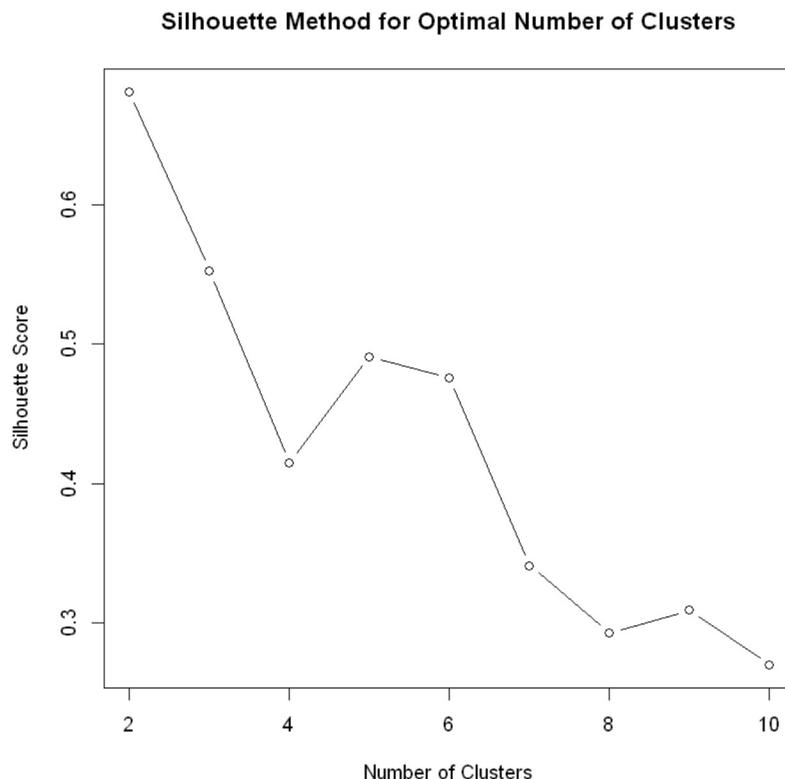
```
library(cluster)
data <- iris[, 1:4]
k_max <- 10
wcss <- numeric(k_max)
for (k in 1:k_max) {
  kmeans_result <- kmeans(data, centers = k)
  wcss[k] <- sum(kmeans_result$withinss)
}
plot(1:k_max, wcss, type = "b", xlab = "Number of Clusters", ylab = "Within-cluster Sum of Squares (WCSS)",
     main = "Elbow Method for Optimal Number of Clusters")
```



In [171...]

```
library(cluster)
library(fpc)
data <- iris[, 1:4]
k_max <- 10
silhouette_scores <- numeric(k_max)

for (k in 2:k_max) {
  kmeans_result <- kmeans(data, centers = k)
  silhouette_scores[k] <- cluster.stats(dist(data), kmeans_result$cluster)$avg.silwidth
}
plot(2:k_max, silhouette_scores[2:k_max], type = "b", xlab = "Number of Clusters", ylab = "Silhouette Score",
     main = "Silhouette Method for Optimal Number of Clusters")
```



In [172...]

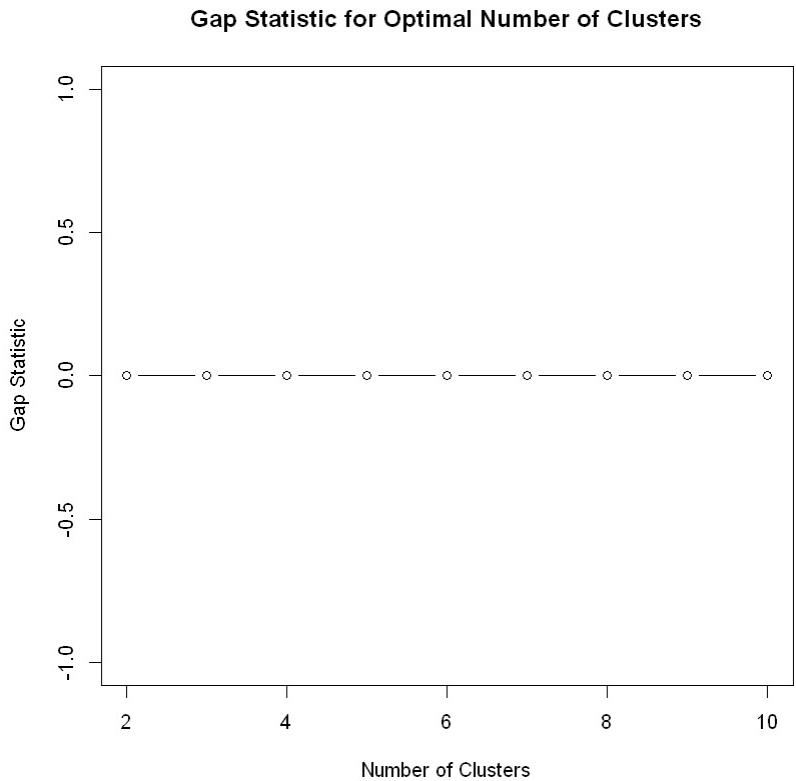
```
library(cluster)
library(fpc)

data <- iris[, 1:4]
k_max <- 10
gap_stat <- numeric(k_max)

for (k in 2:k_max) {
  tryCatch({
    gap_stat[k] <- clusGap(data, FUNcluster = kmeans, K.max = k, B = 50)$Gap[k]
  }, error = function(e) {
    cat("Error occurred for k =", k, "\n")
  })
}

plot(2:k_max, gap_stat[2:k_max], type = "b", xlab = "Number of Clusters", ylab = "Gap Statistic",
      main = "Gap Statistic for Optimal Number of Clusters")
```

Error occurred for k = 2
Error occurred for k = 3
Error occurred for k = 4
Error occurred for k = 5
Error occurred for k = 6
Error occurred for k = 7
Error occurred for k = 8
Error occurred for k = 9
Error occurred for k = 10



END