Problem Statement : Iris Flower Classification

Import Necessary Library

In [4]:

```
library(ggplot2)
```

Load the Dataset

In [5]:

```
df <- iris
```

In [6]:

```
head(df)
```

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

Summary of the dataset

In [14]:

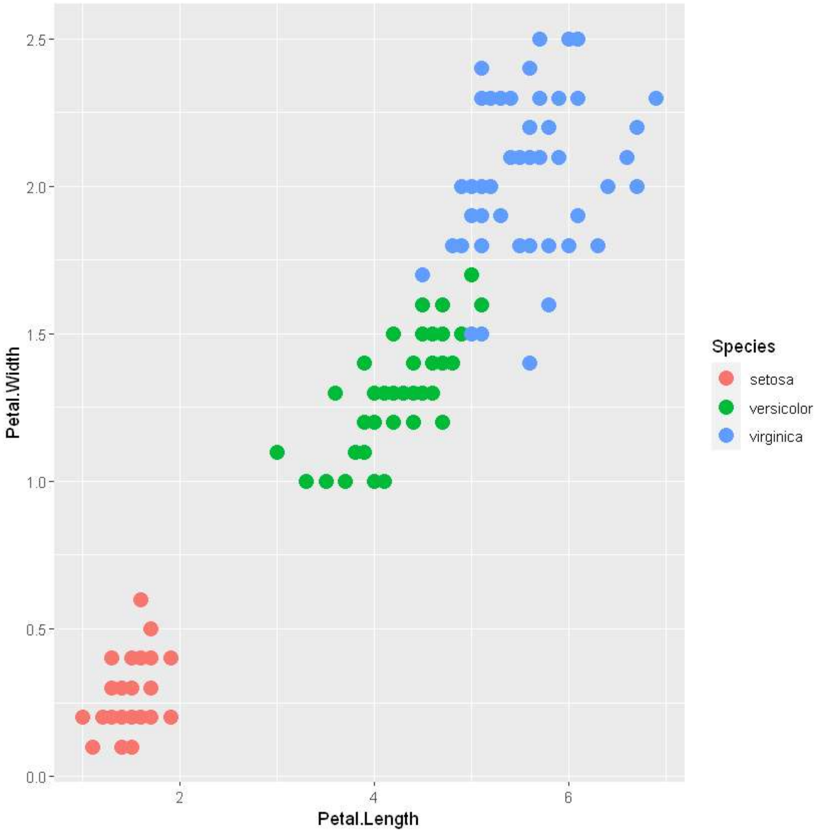
```
summary(df)
```

```
  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  
```

Data Visualization

In [7]:

```
ggplot(df, aes(Petal.Length, Petal.Width)) + geom_point(aes(col=Species), size=4)
```



Eliminating the target variable

In [8]:

```
Xtrain <- df[, -5]

head(Xtrain)
```

A data.frame: 6 × 4

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

Scale the dataset

In [9]:

```
scal <- scale(Xtrain)
```

In [10]:

```
head(scal)
```

A matrix: 6 × 4 of type dbl

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
-0.8976739	1.01560199	-1.335752	-1.311052
-1.1392005	-0.13153881	-1.335752	-1.311052
-1.3807271	0.32731751	-1.392399	-1.311052
-1.5014904	0.09788935	-1.279104	-1.311052
-1.0184372	1.24503015	-1.335752	-1.311052
-0.5353840	1.93331463	-1.165809	-1.048667

In [11]:

```
levels(df$Species)
```

'setosa' 'versicolor' 'virginica'

In [12]:

```
ytrain <- iris[,5]
```

In [13]:

```
ytrain
```

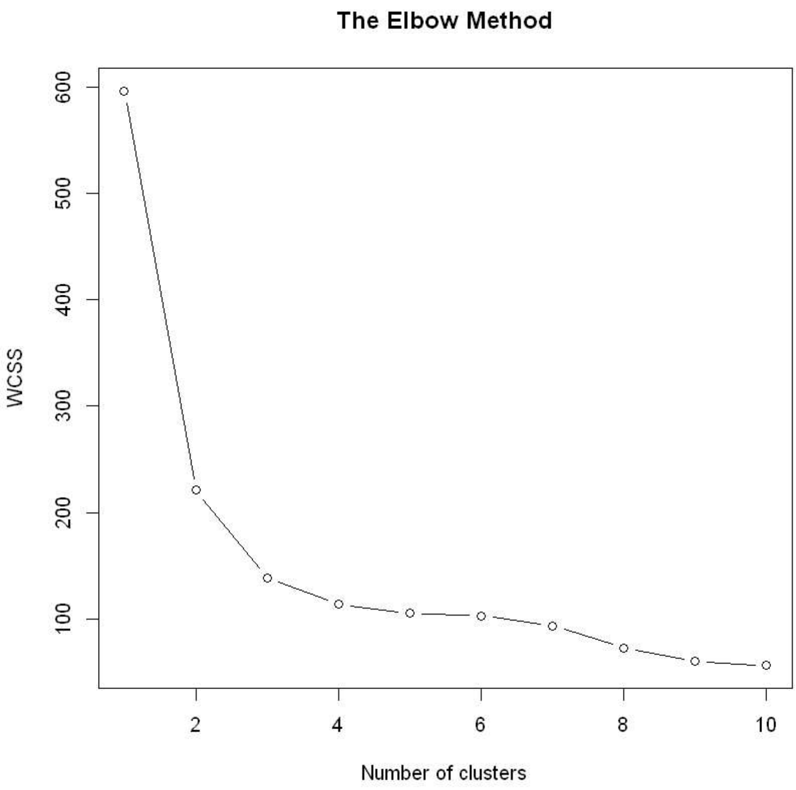
setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa
setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa
setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa
setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa
setosa setosa setosa setosa setosa setosa versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
versicolor versicolor versicolor versicolor versicolor versicolor virginica virginica
virginica virginica virginica virginica virginica virginica virginica virginica virginica
virginica virginica virginica virginica virginica virginica virginica virginica virginica
virginica virginica virginica virginica virginica virginica virginica virginica virginica
virginica virginica virginica virginica virginica virginica virginica virginica virginica
virginica virginica virginica virginica virginica virginica virginica virginica virginica
virginica virginica virginica

▶ **Levels**:

Using the elbow method to find the optimal number of clusters

In [47]:

```
# set.seed(6) #seed - random initialization of clusters
wcss = vector()
maximum <- 10
for (i in 1:maximum) wcss[i] = sum(kmeans(scal, i)$withinss) #withinss: (WSS)Vector of with
plot(1:maximum,
     wcss,
     type = 'b',
     main = paste('The Elbow Method'),
     xlab = 'Number of clusters',
     ylab = 'WCSS')
```



In [ ]:

```
```

For k=3, apply the K-means clustering algorithm.

In [15]:

```
irisCluster <- kmeans(df[,1:4], center=3, nstart=20)
irisCluster
```

```
K-means clustering with 3 clusters of sizes 62, 38, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     6.850000    3.073684     5.742105    2.071053
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1
 [75] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2
2 2
[112] 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2
1 2
[149] 2 1

Within cluster sum of squares by cluster:
[1] 39.82097 23.87947 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withins
s"
[6] "betweenss"    "size"         "iter"         "ifault"
```

In [16]:

```
km <- kmeans(scal[,1:4], center=3, nstart=20)
km
```

```
K-means clustering with 3 clusters of sizes 50, 47, 53

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2   1.13217737  0.08812645    0.9928284   1.0141287
3  -0.05005221 -0.88042696    0.3465767   0.2805873

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 2 3
3 3
 [75] 3 2 2 2 3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2
2 2
[112] 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 2 2 2 2 2 2 3 3 2 2 2 3 2 2 2 3 2 2 2
3 2
[149] 2 3

Within cluster sum of squares by cluster:
[1] 47.35062 47.45019 44.08754
 (between_SS / total_SS =  76.7 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withins
s"
[6] "betweenss"    "size"         "iter"         "ifault"
```

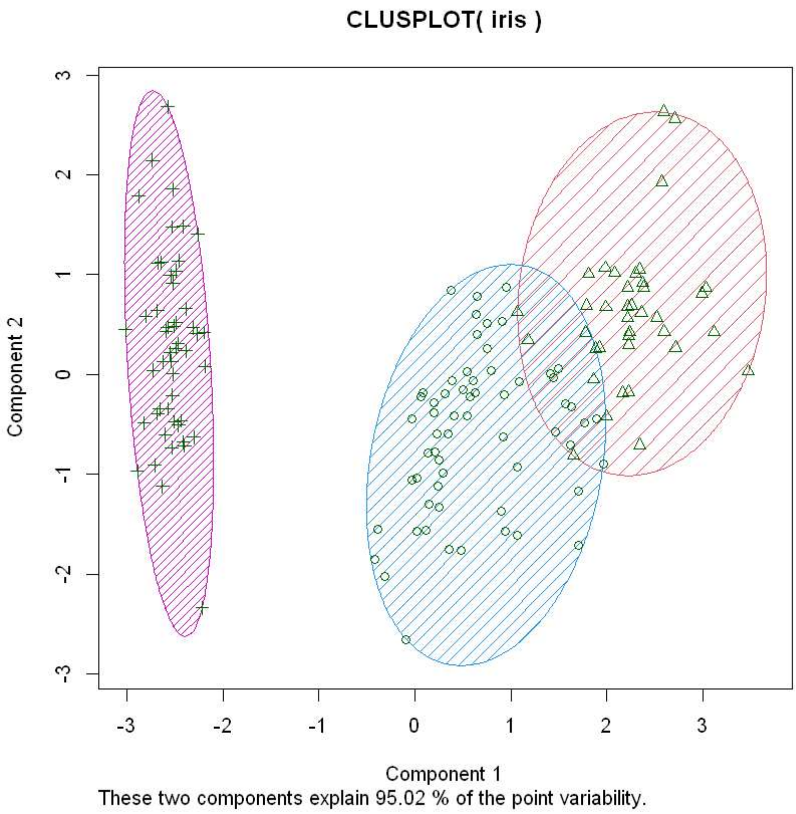
In [17]:

```
table(irisCluster$cluster, df$Species)
```

```
    setosa versicolor virginica
  1      0         48        14
  2      0          2        36
  3     50          0         0
```

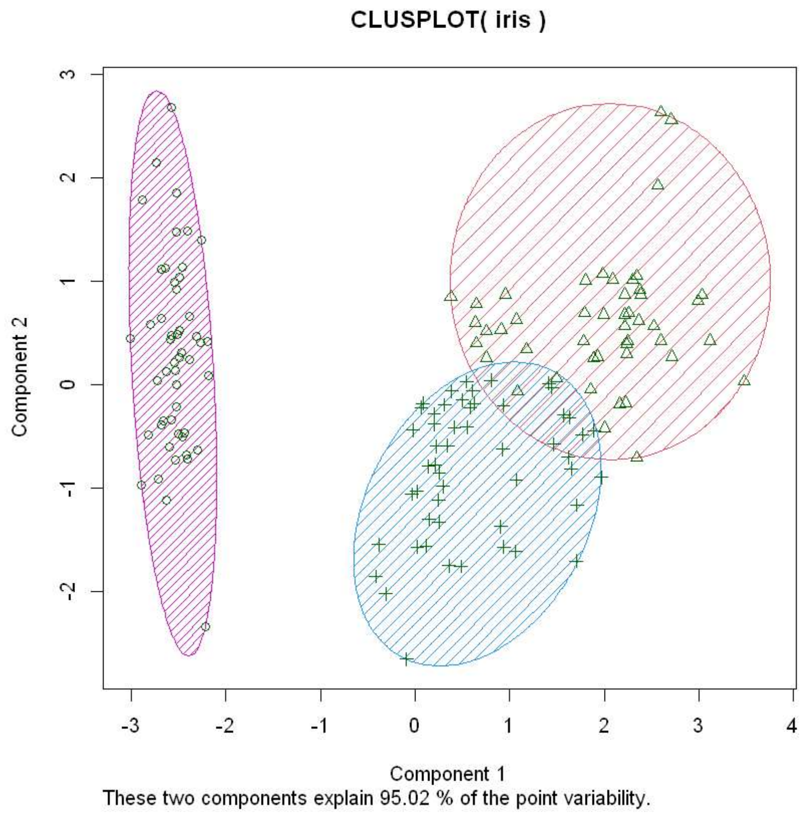
In [18]:

```
library(cluster)
clusplot(iris, irisCluster$cluster, color=T, shade=T, labels=0, lines=0)
```



In [19]:

```
library(cluster)
clusplot(iris, km$cluster, color=T, shade=T, labels=0, lines=0)
```



In [ ]:

In [20]:

```
km$betweenss/km$totss
```

0.766965839400417

In [21]:

```
irisCluster$betweenss/irisCluster$totss
```

0.884275251344648

In [26]:

```
aggregate(Xtrain, by=list(cluster=irisCluster$cluster), mean)
```

A data.frame: 3 × 5

cluster	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	5.901613	2.748387	4.393548	1.433871
2	6.850000	3.073684	5.742105	2.071053
3	5.006000	3.428000	1.462000	0.246000

If you want to add the point classifications to the original data

In [27]:

```
dd <- cbind(Xtrain, cluster = irisCluster$cluster)
head(dd)
```

A data.frame: 6 × 5

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

In [31]:

```
# Cluster number for each of the observations
irisCluster$cluster
```

```
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2
2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 1
```

In [32]:

```
head(irisCluster$cluster, 4)
```

3 3 3 3

In [33]:

```
# Cluster size
irisCluster$size
```

62 38 50

In [34]:

```
# Cluster means
irisCluster$centers
```

A matrix: 3 × 4 of type dbl

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.901613	2.748387	4.393548	1.433871
2	6.850000	3.073684	5.742105	2.071053
3	5.006000	3.428000	1.462000	0.246000

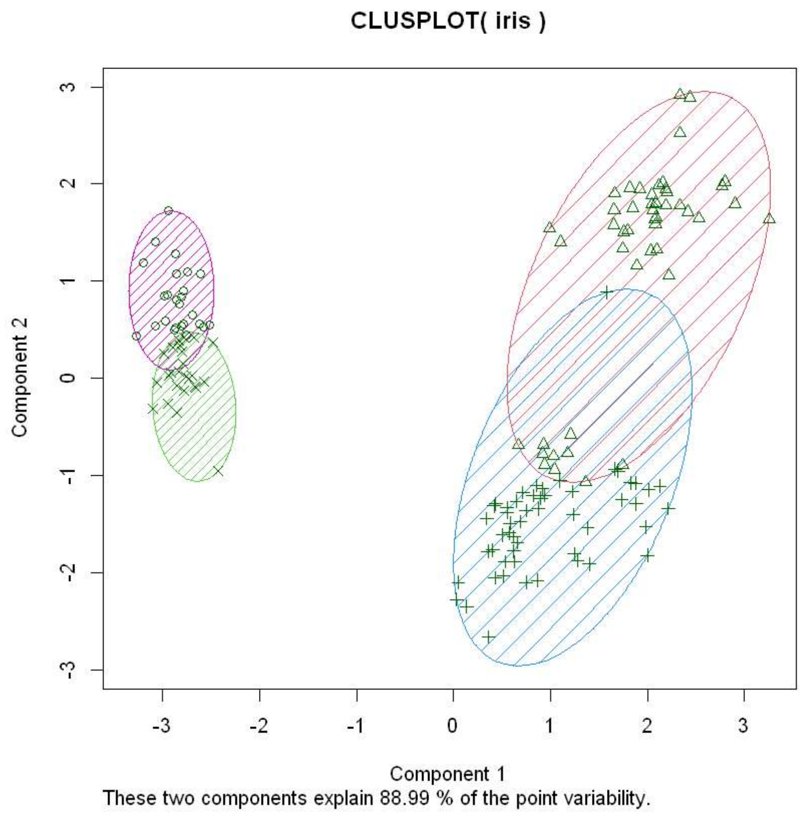
In [50]:

```
km2 <- kmeans(scal[,1:4], center=4, nstart=20)
km2
```

```
K-means clustering with 4 clusters of sizes 25, 47, 53, 25

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -0.71894419  1.50198969   -1.2972312  -1.2165934
2   1.13217737  0.08812645    0.9928284   1.0141287
3  -0.05005221 -0.88042696    0.3465767   0.2805873
4  -1.30343857  0.19883774   -1.3040289  -1.2848136

Clustering vector:
  [1] 1 4 4 4 1 1 4 4 4 4 1 4 4 4 1 1 1 1 1 1 1 1 1 4 4 4 4 1 1 4 4 1 1 1 4
4 1
 [38] 1 4 4 1 4 4 1 1 4 1 4 1 4 2 2 2 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 2 3
3 3
 [75] 3 2 2 2 3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2
2 2
[112] 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 2 2 2 2 2 2 3 3 2 2 2 3 2 2 2 3 2 2 2
3 2
[149] 2 3

Within cluster sum of squares by cluster:
[1] 12.147537 47.450194 44.087545  9.646348
 (between_SS / total_SS =  81.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withins
s"
[6] "betweenss"    "size"         "iter"         "ifault"
```

In [51]:

```
clusplot(iris, km2$cluster, color=T, shade=T, labels=0, lines=0)
```



In [52]:

```
km2$tot.withinss
```

113.331623512778

In [ ]:

```

```