Comparison of Classification and Clustering Algorithms on Iris Dataset Using R

Talha Hanif Butt

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(mclust)
```

```
## Package 'mclust' version 5.4.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(fpc)
library(cluster)
library(clusteval)
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(ggplot2)
library(kmed)
```

Loading Iris Dataset

```
# attach the iris dataset to the environment
data(iris)
# rename the dataset
dataset <- iris
```

Partitioning Data for Validation

```
# create a list of 80% of the rows inthe original dataset we can use for training
validation_index <- createDataPartition(dataset$Species, p=0.80, list=FALSE)
# select 20% of the data for validation
validation <- dataset[-validation_index,]
# use the remaining 80% of data to training and testing the models
dataset <- dataset[validation_index,]
```

Getting Insights from Data

```
# dimensions of dataset
dim(dataset)
```

```
## [1] 120   5
```

```
# list types for each attribute
sapply(dataset, class)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##    "numeric"    "numeric"    "numeric"    "numeric"     "factor"
```

```
# take a peek at the first 5 rows of the data
head(dataset)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 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
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
## 7          4.6         3.4          1.4         0.3  setosa
```

```
# list the levels for the class
levels(dataset$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

```
# summarize the class distribution
percentage <- prop.table(table(dataset$Species)) * 100
cbind(freq=table(dataset$Species), percentage=percentage)
```

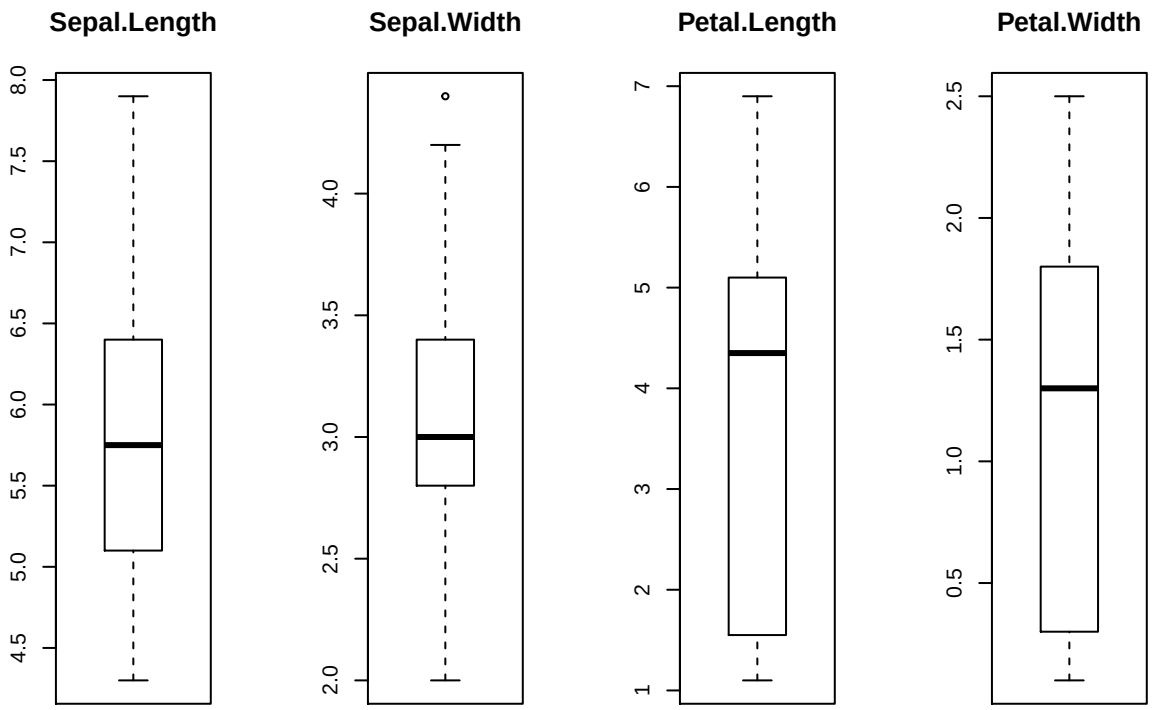
```
##            freq percentage
## setosa       40   33.33333
## versicolor   40   33.33333
## virginica    40   33.33333
```

```
# summarize attribute distributions
summary(dataset)
```

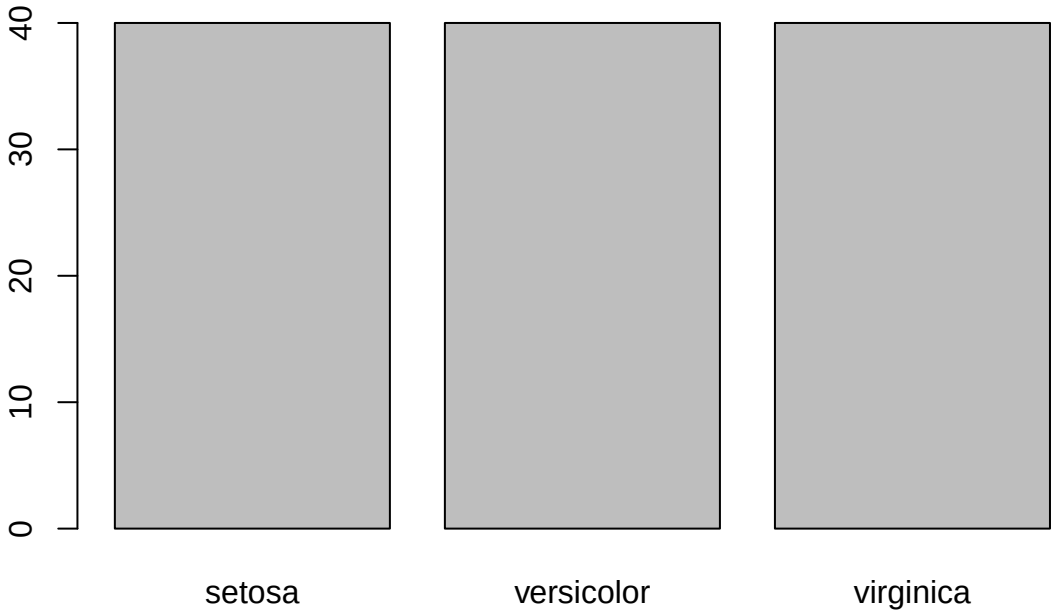
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.100   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.575   1st Qu.:0.300
##  Median :5.750   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.838   Mean   :3.059   Mean   :3.763   Mean   :1.183
##  3rd Qu.:6.400   3rd Qu.:3.400   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :40
##  versicolor:40
##  virginica :40
##
##
##
```

```
# split input and output
x <- dataset[,1:4]
y <- dataset[,5]
```

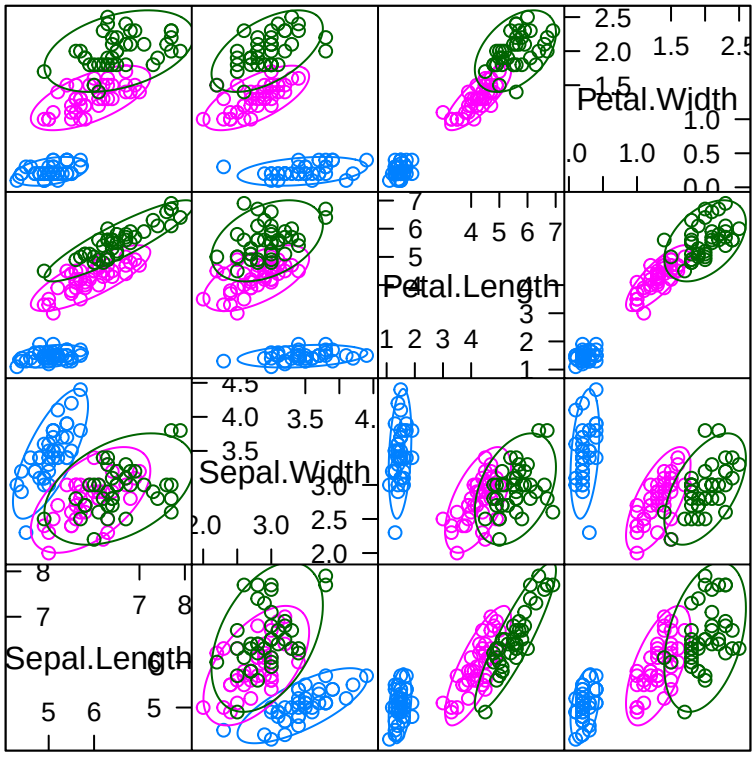
```
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 1:4) {
  boxplot(x[,i], main=names(iris)[i])
}
```




```
# barplot for class breakdown
plot(y)
```

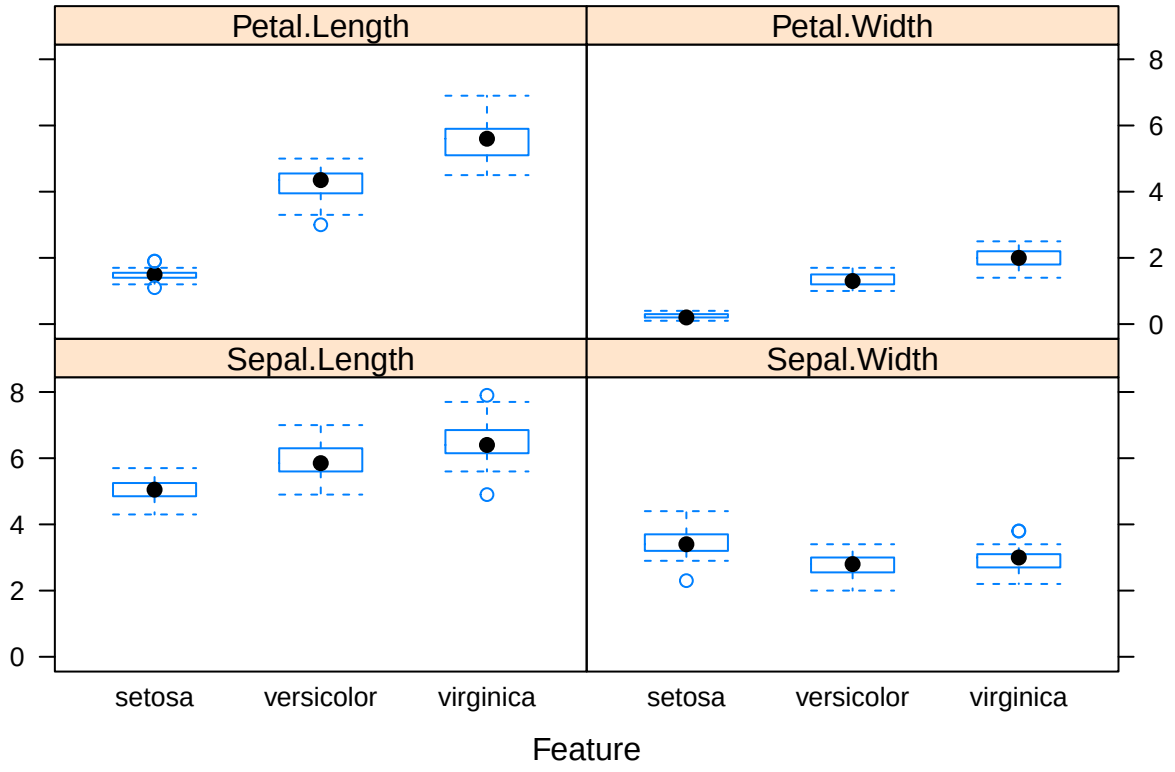



```
# scatterplot matrix
featurePlot(x=x, y=y, plot="ellipse")
```

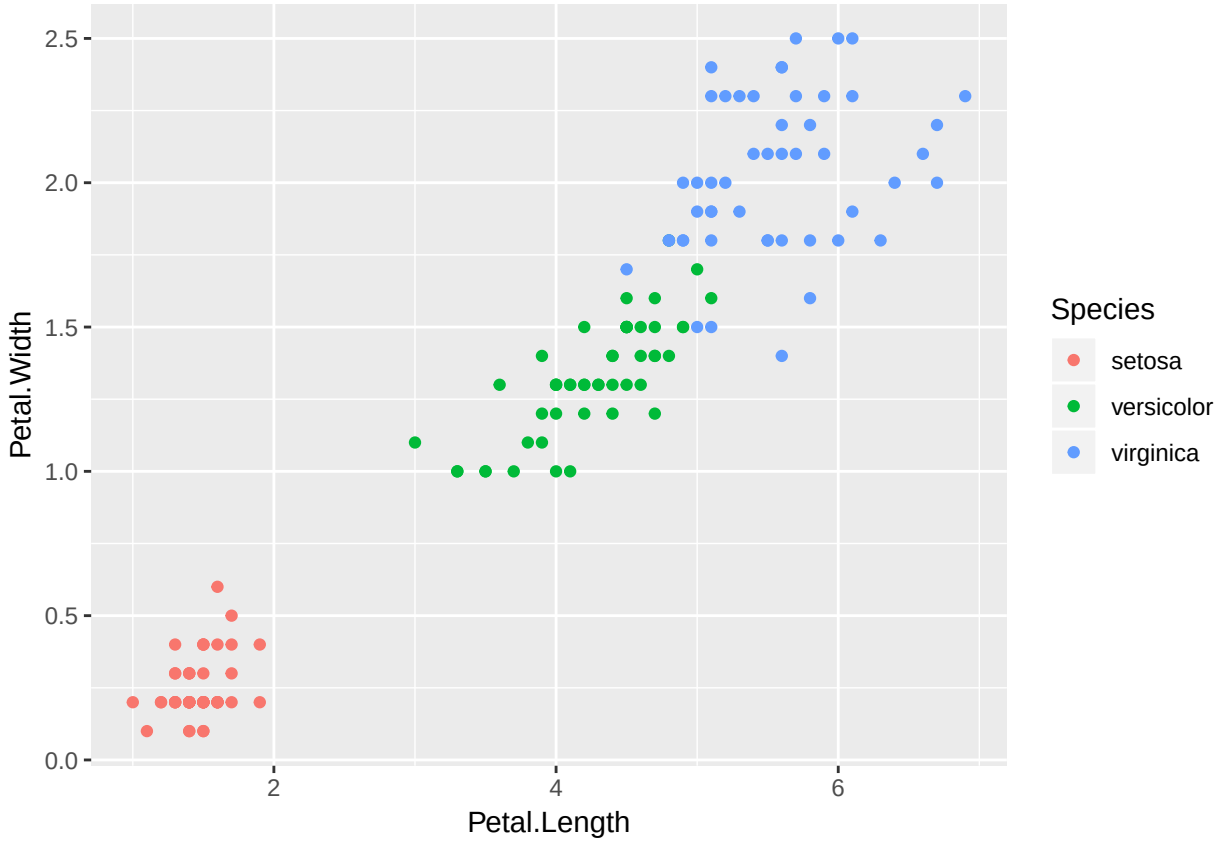


Scatter Plot Matrix

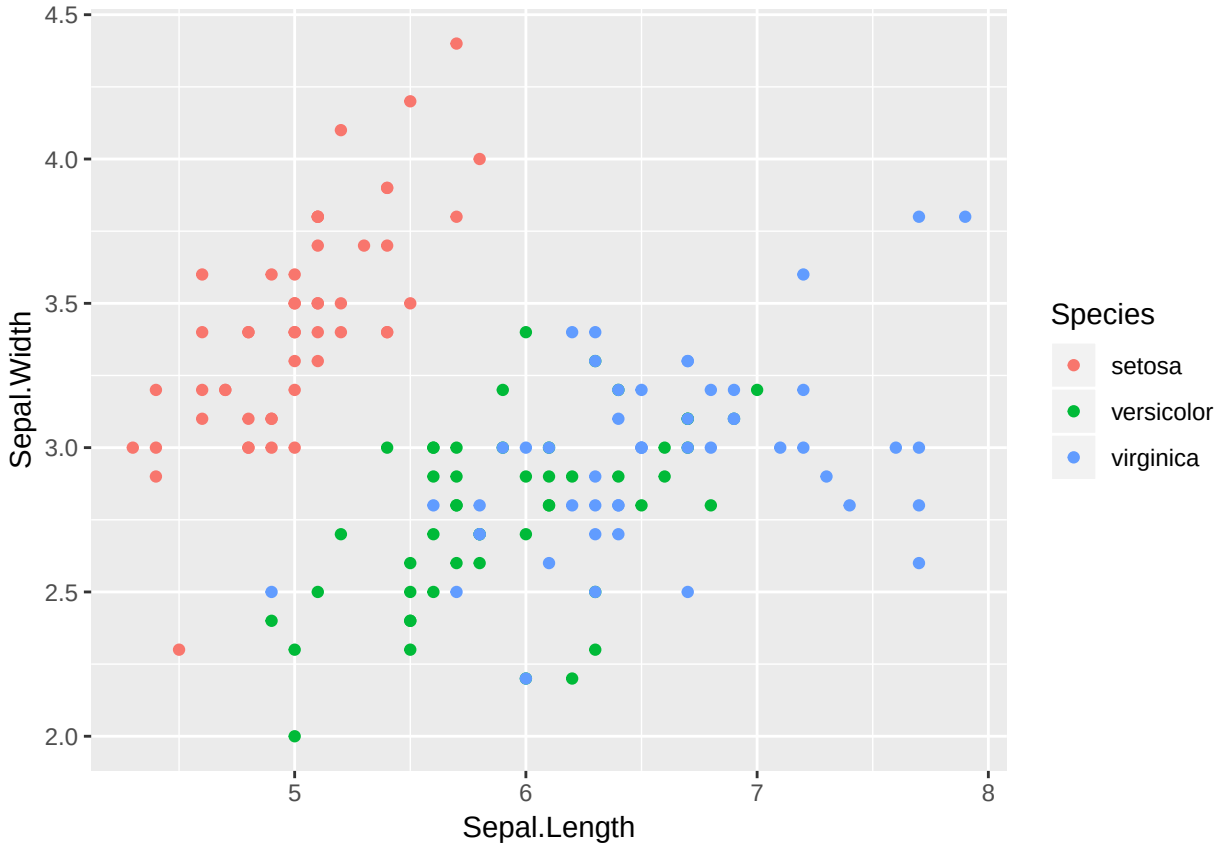
```
# box and whisker plots for each attribute
featurePlot(x=x, y=y, plot="box")
```



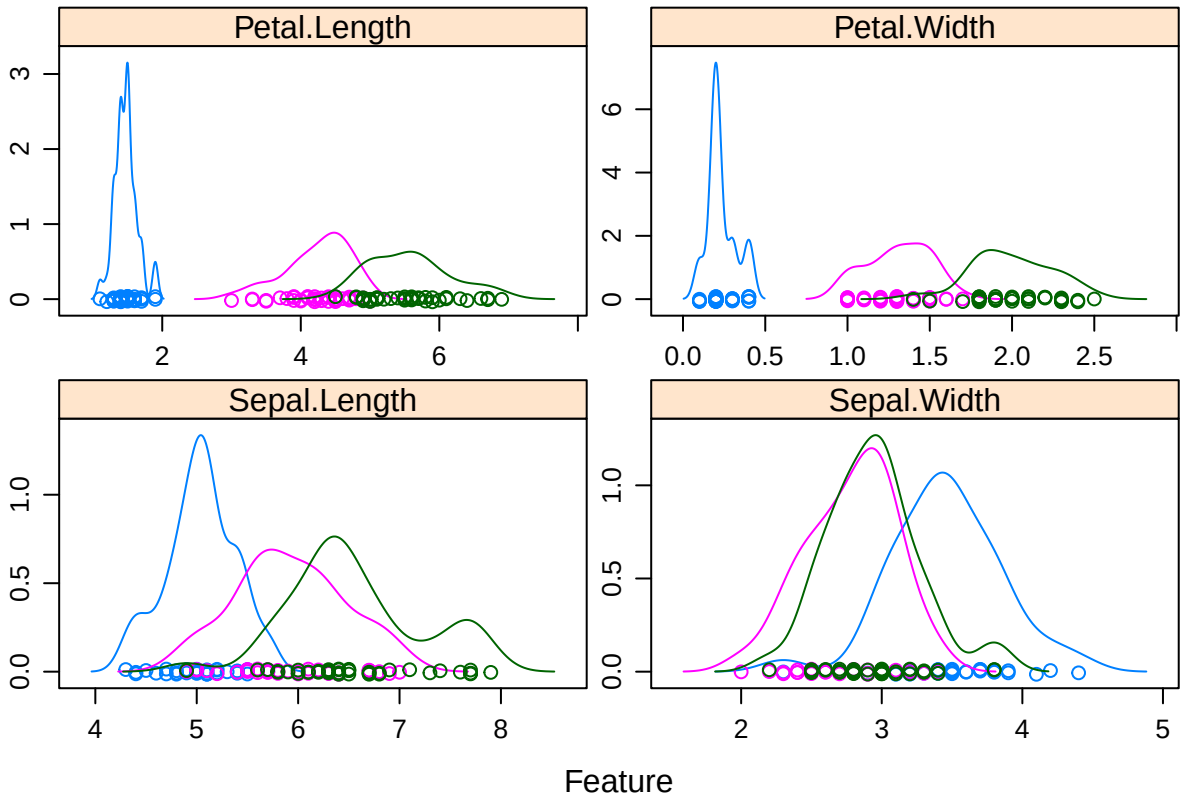
```
ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom_point()
```




```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) + geom_point()
```




```
# density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```



Applying Classification Algorithms on Iris Data

```
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

```
# kNN
set.seed(7)
fit.knn <- train(Species~., data=dataset, method="knn", metric=metric, trControl=control)
# SVM
set.seed(7)
fit.svm <- train(Species~., data=dataset, method="svmRadial", metric=metric, trControl=control)
# Random Forest
set.seed(7)
fit.rf <- train(Species~., data=dataset, method="rf", metric=metric, trControl=control)
```

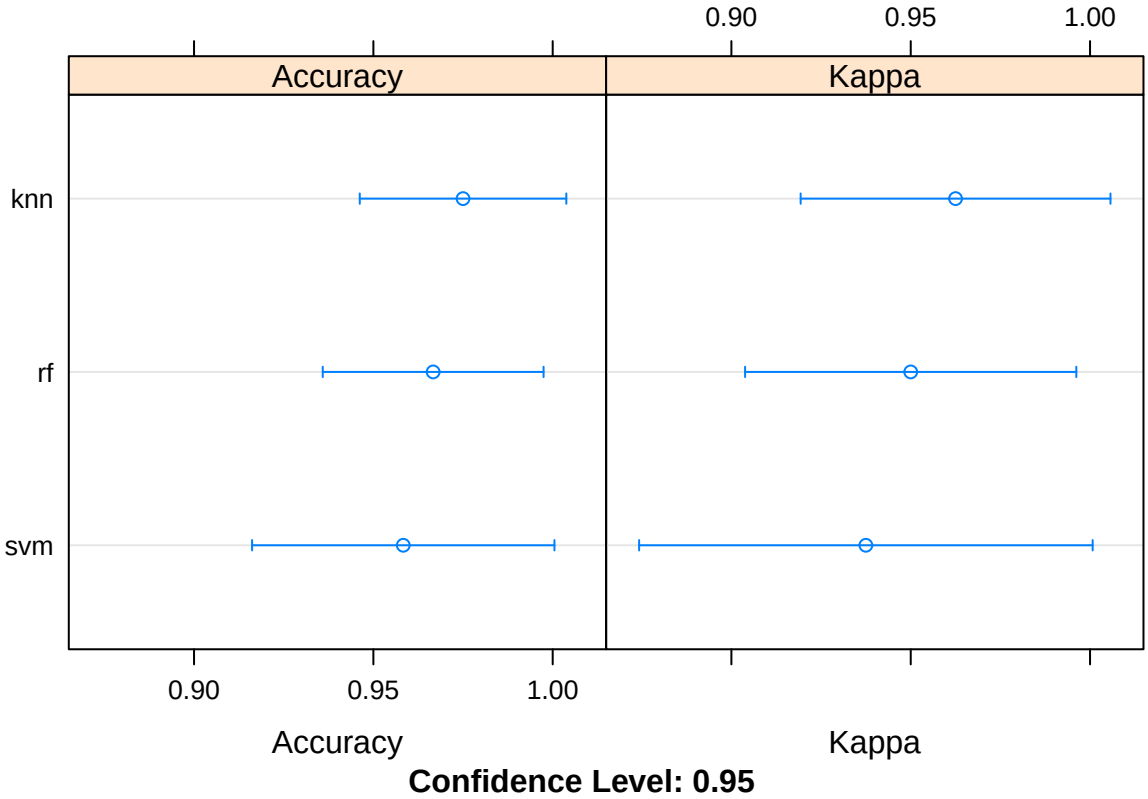
Comparison of the Classification Algorithms

```
# summarize accuracy of models
results <- resamples(list(knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
```

```
## Models: knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu. Median      Mean 3rd Qu. Max. NA's
## knn 0.9166667 0.9375000      1 0.9750000       1    1    0
## svm 0.8333333 0.9166667      1 0.9583333       1    1    0
## rf  0.9166667 0.9166667      1 0.9666667       1    1    0
##
## Kappa
##      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## knn 0.875 0.90625      1 0.9625       1    1    0
## svm 0.750 0.87500      1 0.9375       1    1    0
## rf  0.875 0.87500      1 0.9500       1    1    0
```

```
# compare accuracy of models
dotplot(results)
```



Insights from the best model

```
# summarize Best Model
print(fit.knn)
```

```
## k-Nearest Neighbors
##
## 120 samples
##   4 predictors
##   3 classes: 'setosa', 'versicolor', 'virginica'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 108, 108, 108, 108, 108, 108, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy  Kappa
##   5  0.975     0.9625
##   7  0.975     0.9625
##   9  0.975     0.9625
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
# estimate skill of KNN on the validation dataset
predictions <- predict(fit.knn, validation)
confusionMatrix(predictions, validation$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0          8         0
##   virginica       0          2        10
##
## Overall Statistics
##
##                Accuracy : 0.9333
##                  95% CI : (0.7793, 0.9918)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 8.747e-12
##
##                   Kappa : 0.9
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.8000           1.0000
## Specificity                 1.0000            1.0000           0.9000
## Pos Pred Value              1.0000            1.0000           0.8333
## Neg Pred Value              1.0000            0.9091           1.0000
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.2667           0.3333
## Detection Prevalence        0.3333            0.2667           0.4000
## Balanced Accuracy           1.0000            0.9000           0.9500
```

Applying Clustering Algorithms

```
# K-means
set.seed(20)
fit.kmeans <- kmeans(iris[, 0:4], 3, nstart = 20)
```

```r
# Hierarchical Agglomerative
set.seed(20)
d <- dist(iris[,0:4], method = "euclidean") # distance matrix
fit.ha <- hclust(d, method="ward.D")
# K-Medoids Clustering
num <- as.matrix(iris[,0:4])
mrwdist <- distNumeric(num, num, method = "mrw")
fit.kmedoids <- fastkmed(mrwdist, ncluster = 3, iterate = 50)
```

Getting insights from Hierarchical Agglomerative Clustering

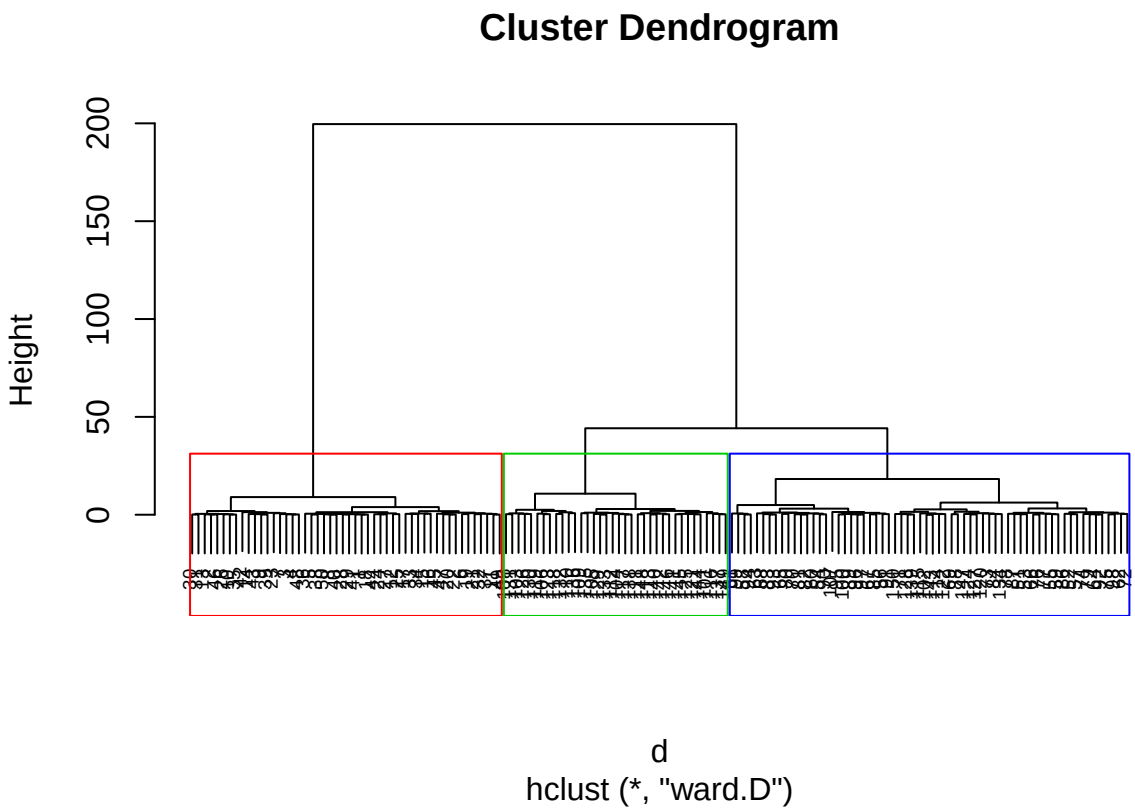
```r
# Cut tree into 4 groups
sub_grp <- cutree(fit.ha, k = 3)

# Number of members in each cluster
table(sub_grp)
```

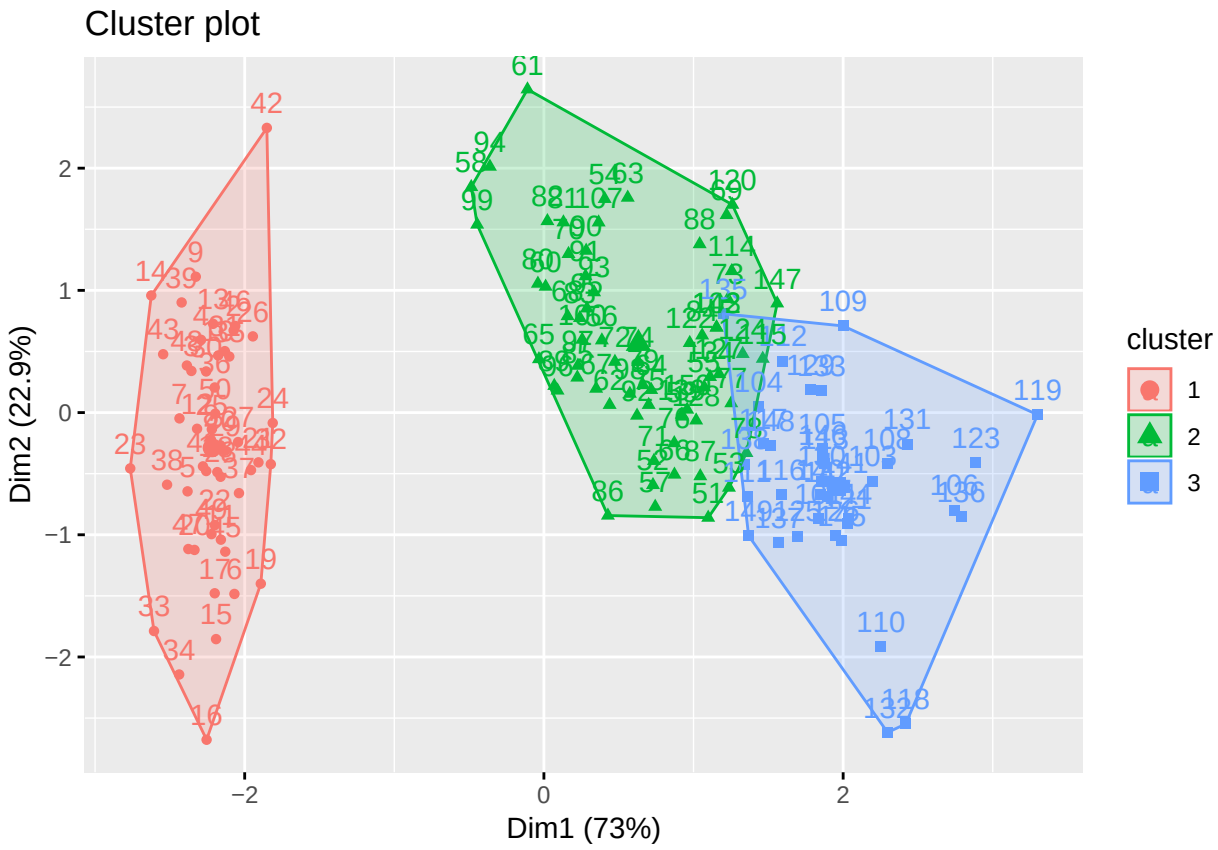
```
## sub_grp
##  1  2  3
## 50 64 36
```

```
## sub_grp
```

```r
plot(fit.ha, cex = 0.6)
rect.hclust(fit.ha, k = 3, border = 2:5)
```




```r
fviz_cluster(list(data = iris[,0:4], cluster = sub_grp))
```

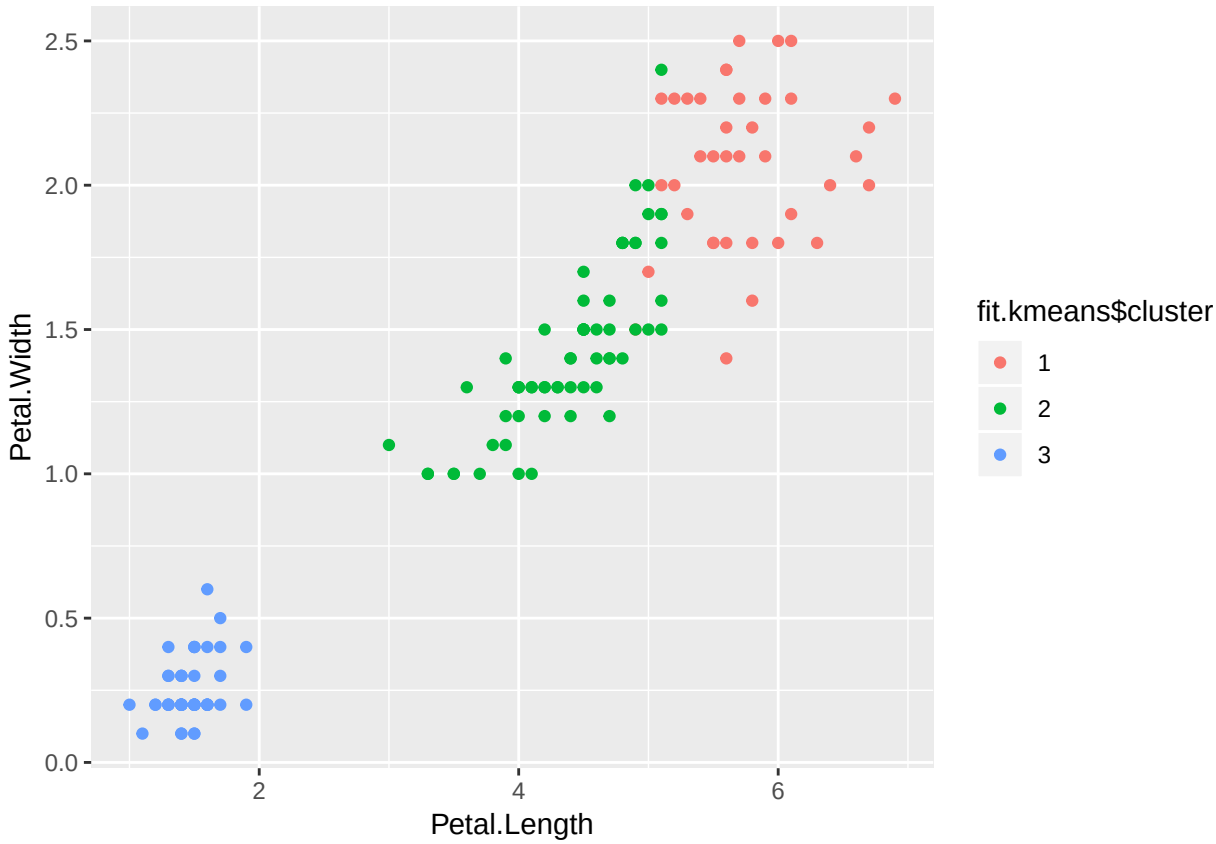


Getting insights from K-Means Clustering

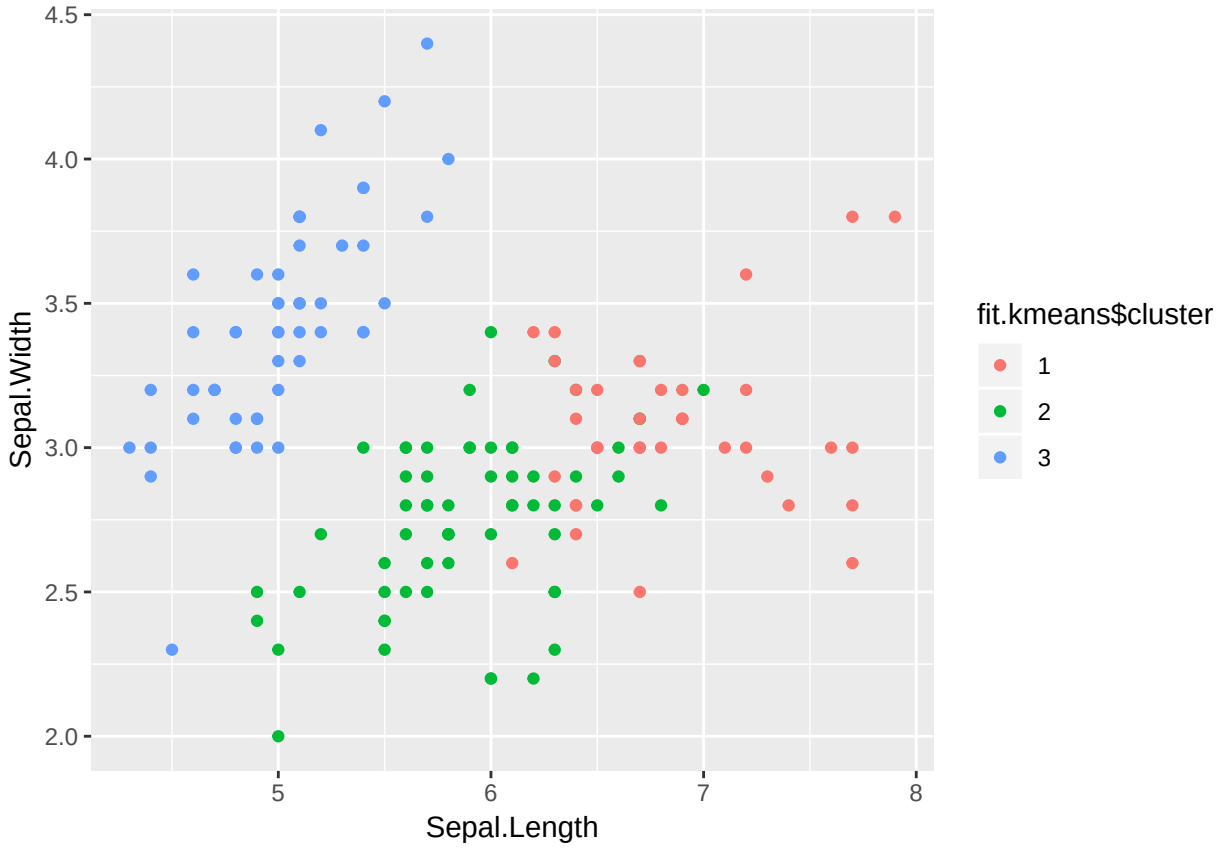
```
table(fit.kmeans$cluster, iris$Species)
```

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

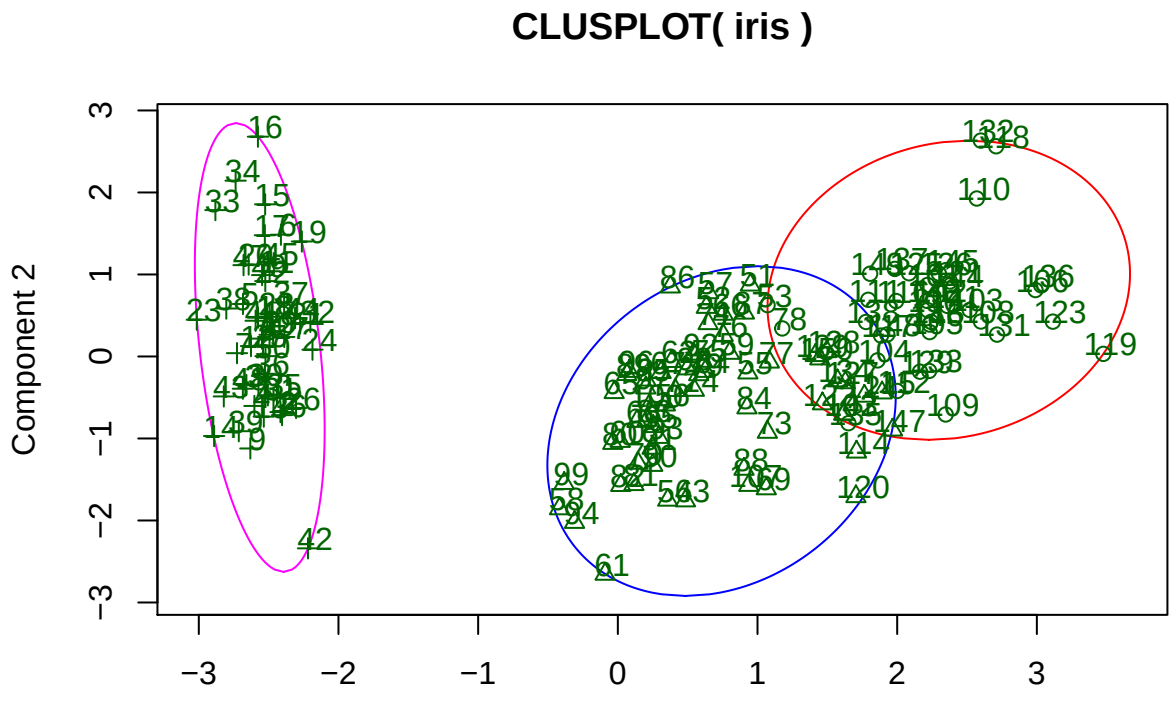
```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(iris, aes(Petal.Length, Petal.Width, color = fit.kmeans$cluster)) + geom_point()
```




```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = fit.kmeans$cluster)) + geom_point()
```




```
clusplot(iris, fit.kmeans$cluster, color=TRUE, shade=FALSE, labels=3, lines=0)
```



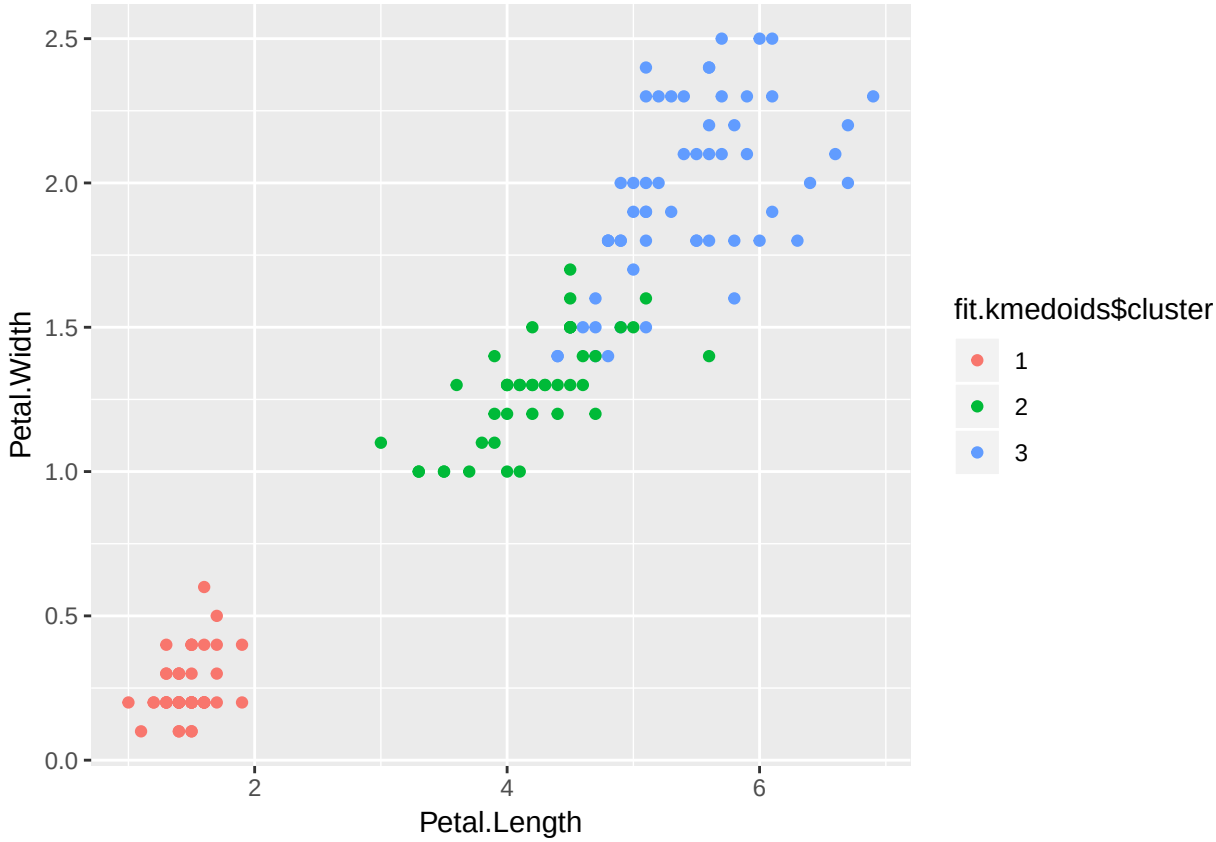
These two components explain 95.02 % of the point variability.

Getting insights from K-Medoids Clustering

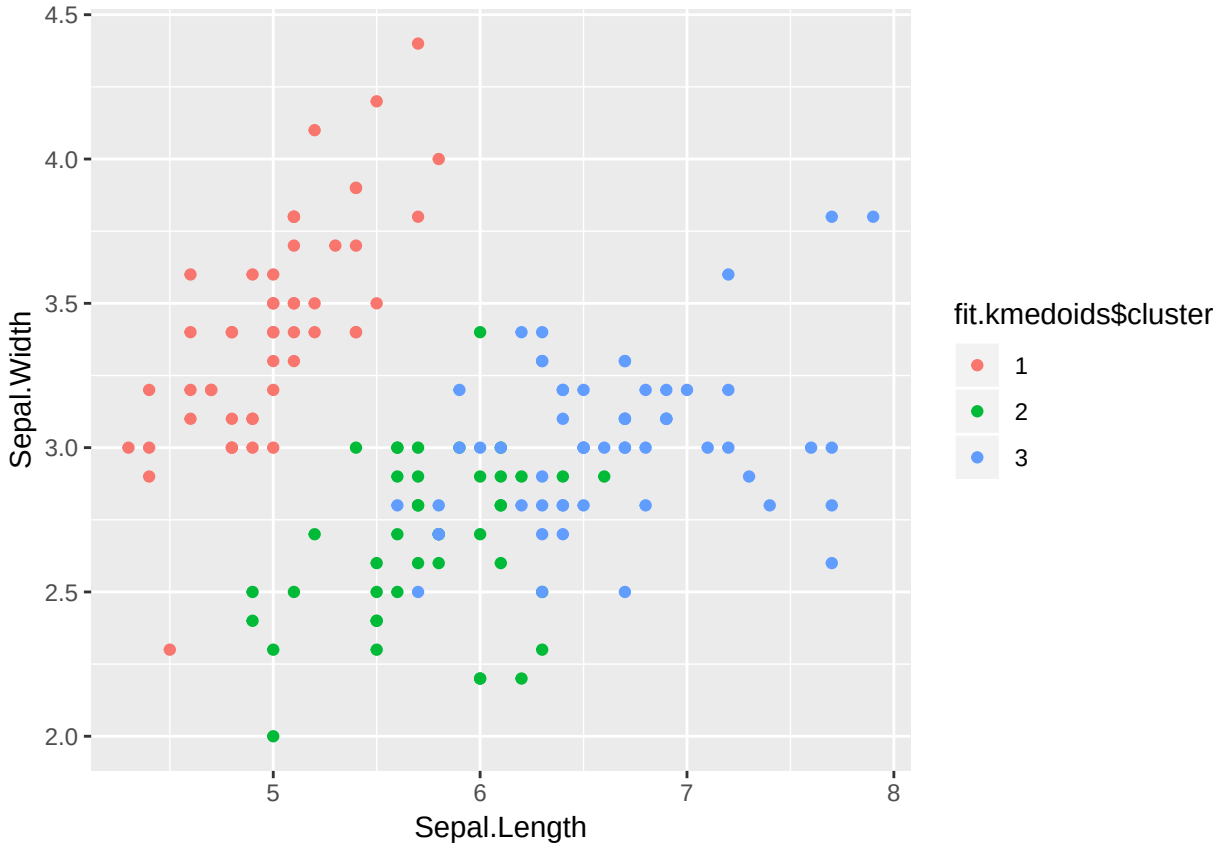
```
table(fit.kmedoids$cluster, iris[,5])
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         39         3
##   3      0         11        47
```

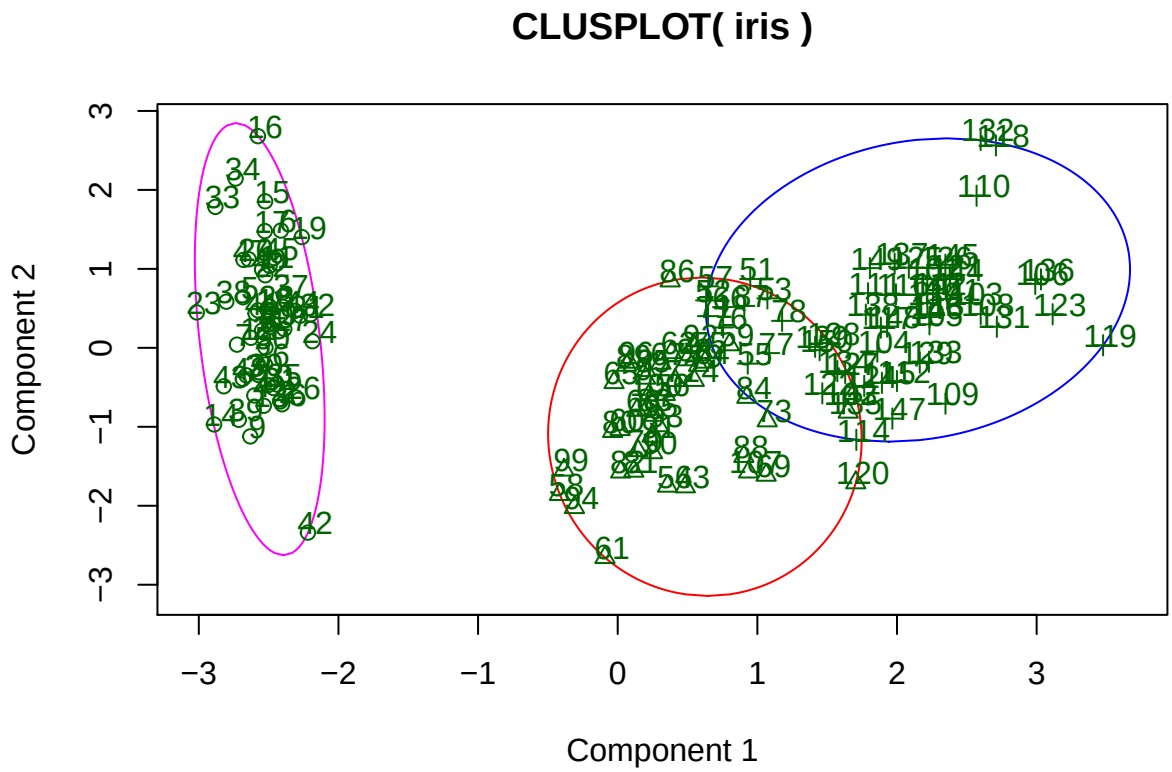
```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(iris, aes(Petal.Length, Petal.Width, color = fit.kmedoids$cluster)) + geom_point()
```




```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = fit.kmedoids$cluster)) + geom_point()
```




```
clusplot(iris, fit.kmedoids$cluster, color=TRUE, shade=FALSE, labels=3, lines=0)
```



These two components explain 95.02 % of the point variability.

Conclusion

With better accuracy and kappa measures, KNN has outperformed other competitors on Iris Dataset while Hierarchical Agglomerative Clustering is the winner when compared with K-Means and K-Medoids Clustering on Iris Dataset as it has clustered data better evident from the Cluster Plot and Cluster Dendrogram.