MIS 545 – Clustering Analysis
Predict Flower Species based on Petal and Sepal Measurements
Kirsten Fure – August 8, 2019

```
if(!require(ggplot2)){
install.packages("ggplot2")
}
library(ggplot2)
if(!require(animation)){
install.packages("animation")
}
 library(animation)
if(!require(fpc)){
install.packages("fpc")
}
library(fpc)

setwd("  ")
iris_full <- read.csv("iris.csv")
head(iris_full)
iris = iris_full[ , c('Petal.Length', 'Petal.Width')]
head(iris)
plot(iris)
normIt <- function(feature){
                normalized <- ((feature - min(feature)) /
 (max(feature) - min(feature)))
                return (normalized)
        }
nor_iris <- apply(iris[,c(1,2)], 2, FUN = normIt)
nor_iris <- as.data.frame(nor_iris)
head(nor_iris)
K5 <- kmeans(nor_iris, 5)
class(K5)
str(K5)
kmeans.totwithinss.k <- function(dataset, number_of_centers){
                km <- kmeans(dataset, number_of_centers)
                km$tot.withinss
        }
c2 = kmeans.totwithinss.k(nor_iris, 2)
c2
c3 = kmeans.totwithinss.k(nor_iris, 3)
c3
c4 = kmeans.totwithinss.k(nor_iris, 4)
```

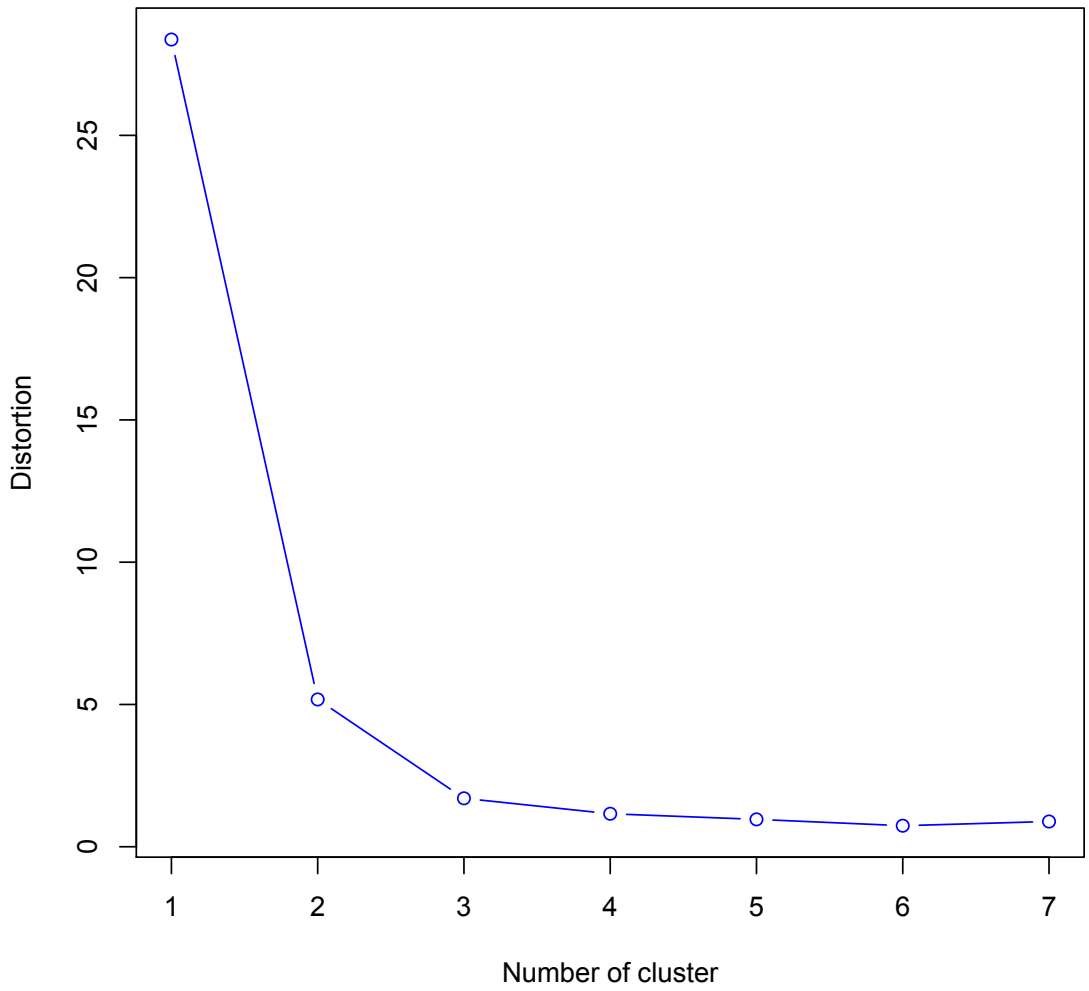
```r
c4
c5 = kmeans.totwithinss.k(nor_iris, 5)
c5
c6 = kmeans.totwithinss.k(nor_iris, 6)
c6

kmeans.distortion <- function(dataset, maxk){
              vec <- as.vector(1:maxk)
               vec[1:maxk] <- sapply(1:maxk, kmeans.totwithinss.k,
           dataset = dataset)
           return (vec)
        }
maxk <- 7
dis_vct <- kmeans.distortion(nor_iris, maxk)
#   Elbow Curve
plot(1:maxk,
      dis_vct,
      type = 'b',
      col = 'blue',
      xlab = "Number of cluster",
      ylab = "Distortion"
        )

num_cluster = 3
#animation of k-means
result <- kmeans.ani(nor_iris, num_cluster)

# result$centers contains average geo-location, which are the
 centers for each clusters.
# The second aggregate method counts the number of points in each
 cluster
centers <- as.data.frame(result$centers)
centers
counts <- aggregate(nor_iris, by = list(result$cluster), FUN =
 length)[,2]
counts
nor_iris$cluster <- result$cluster
head(nor_iris)

# attach cluster label based on the distance from center of
 clusters to position(0,0)
# k-means is a clustering algorithm, assuming a isotropic data as
 input
ra <- as.data.frame(centers$Petal.Length^2 +
```

```
 centers$Petal.Width^2)
ra
ra$real <- rank(ra)  #sort of their distance from origin
nominal <- c()
for(i in (1:nrow(ra))){
nominal[i] <- i
    }
ra$nominal <- nominal    #I think nominal is the row number?
# retag cluster number
result$real <- c()
result
for (i in (1:length(result$cluster))){
    for(j in (1:4)){
        if(result$cluster[i] == j){
        result$real[i] <- ra$real[j]
    }
  }
}
result

# confusion matrix
table(result$real, nor_iris$cluster)    #reassinging cluster
 number based on distance from origina
head(nor_iris)

 # base layer
 plot.iris <- ggplot(data = nor_iris, aes(x = Petal.Length, y =
Petal.Width, color = result$cluster))
 plot.iris

 # alpha: semi-transparent points
 plot.iris + geom_point(alpha = .25, size = 5) +
                # cluster centers, colored black:
                geom_point(data = centers, aes(x = Petal.Length,
 y = Petal.Width), size = 5, color = 'black')  +
                # cool colors for each cluster:
                scale_color_gradientn(colours =
 rainbow(num_cluster)) +
                # add a title, align to the center
                theme(plot.title = element_text(hjust = 0.5)) +
                ggtitle("GGPlot of K-means clusters")


#   eps:     Reachability distance
```

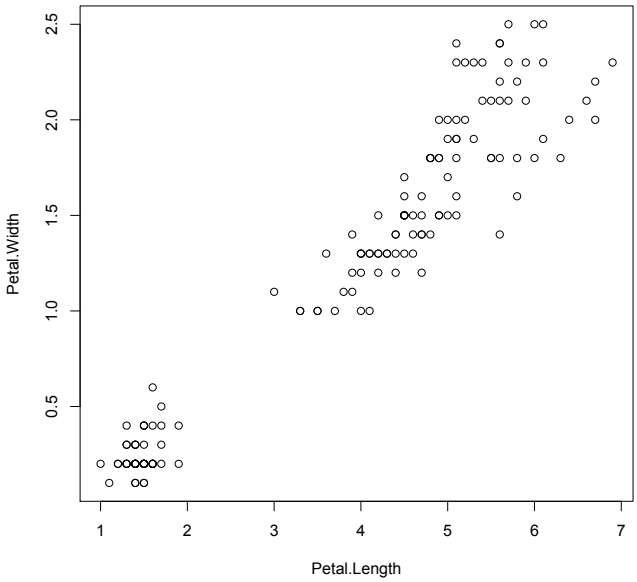
```
#   MinPts:  Reachability minimum no. of points,
#   scale:   Scale the data if TRUE

ds <- dbscan(nor_iris[,c(1,2)], eps = .4, MinPts = 4, scale =
 TRUE, showplot = 1, seeds = TRUE, method = "hybrid")
ds

# triangle:            core points
# circle in color:     border points
# else:                outliers or noises
#   confusion matrix
table(ds$cluster, nor_iris$cluster)   #dbscan is only showing 2
 clusters vs 3

ds$isseed
nor_iris$core <- ds$isseed
nor_iris$outliers <- ds$cluster
nor_iris$core
nor_iris$outliers
head(nor_iris)

 # base layer with DBScan results
plot.dbiris <- ggplot(data = nor_iris, aes(x = Petal.Length, y =
 Petal.Width, color = ds$cluster, shape = nor_iris$core ))

# alpha: semi-transparent points with DBScan results
plot.dbiris + geom_point(alpha = .25, size = 5) +
                # outliers, shape outline black:
                geom_point(data = subset(nor_iris,
 nor_iris$outliers == 0), aes(x = Petal.Length, y = Petal.Width),
 size = 5, color = 'black', shape = 1)  +
                # cool colors for each cluster:
                scale_color_gradientn(colours =
 rainbow(length(unique(ds$cluster)))) +
                 # colour backgound as 'grey'
                theme(panel.background = element_rect(fill =
 'grey', colour = 'black')) +
                # add a title, align to the center
                theme(plot.title = element_text(hjust = 0.5)) +
                ggtitle("GGPLOT DBSCAN")

g=ggplot(iris_full, aes(x=Petal.Length, y=Petal.Width, col =
 Species)) + geom_point()
g = g+ggtitle("Petal Clusters Predict Species")
```

```
plot(g)
```

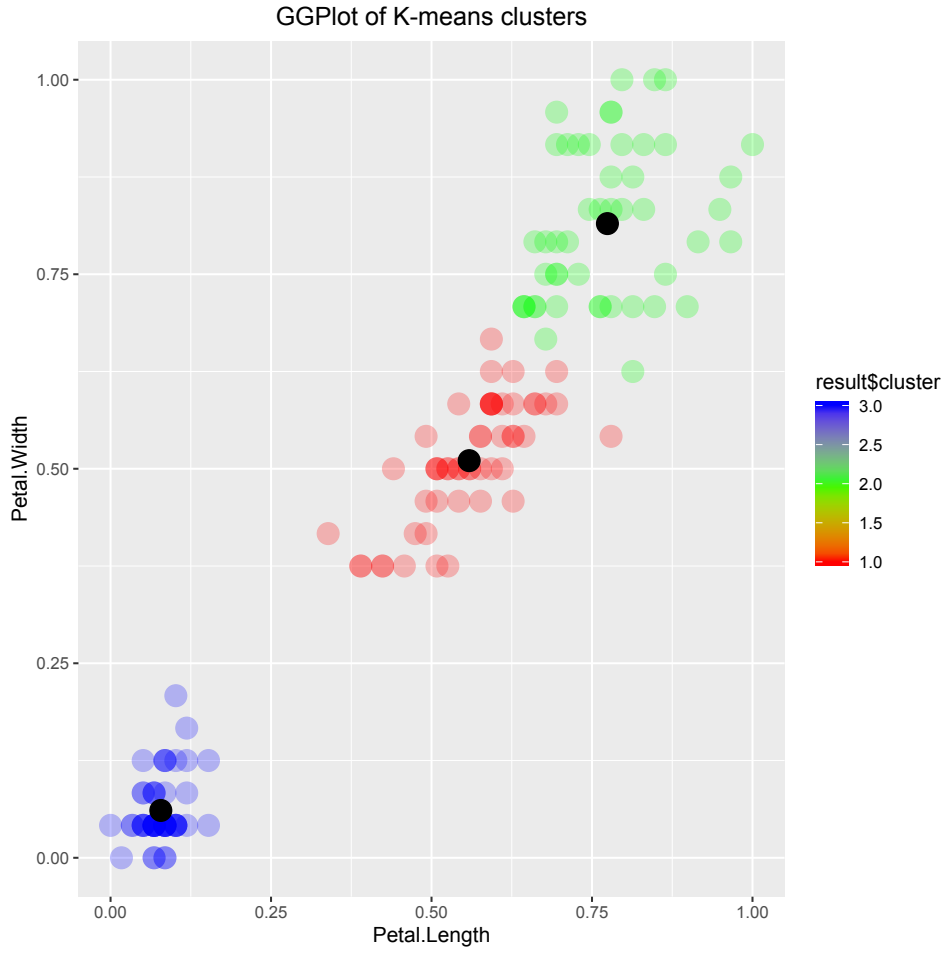
2. This elbow curve shows that K=3 is at the elbow of the plot, so 3 clusters should be a good value for K. Two clusters should also give decent results, as the angle at K=2 is dramatically decreased in comparison with K=1.



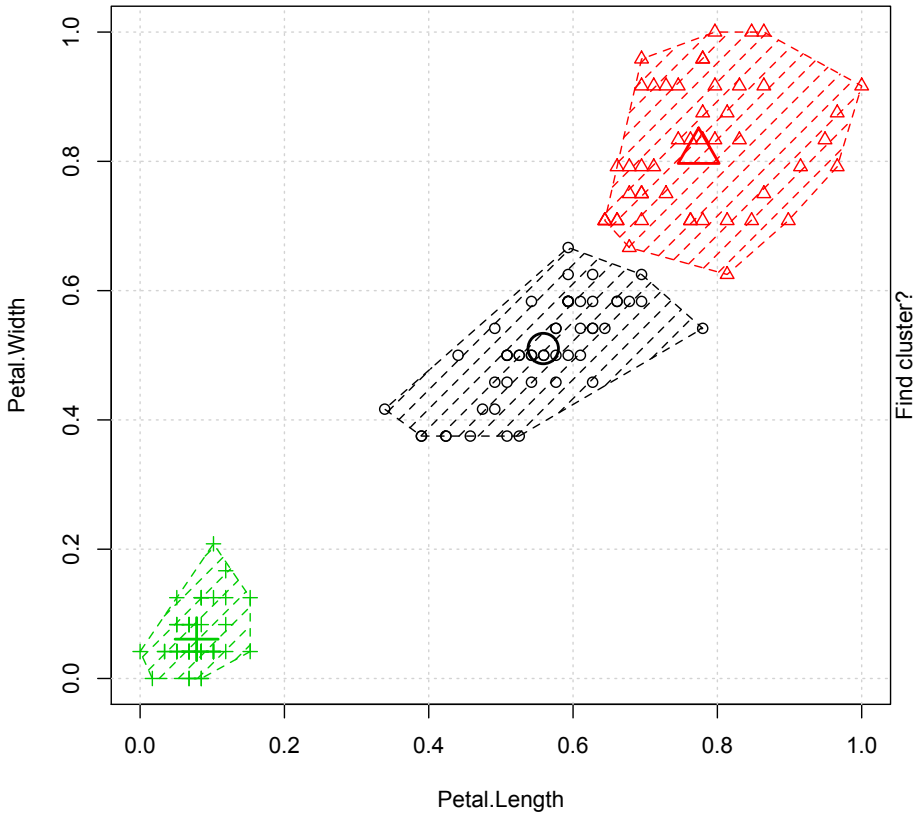
3. Basic scatter plot of Petal.Width vs. Petal.Length, showing that Petal Length and Width are correlated.



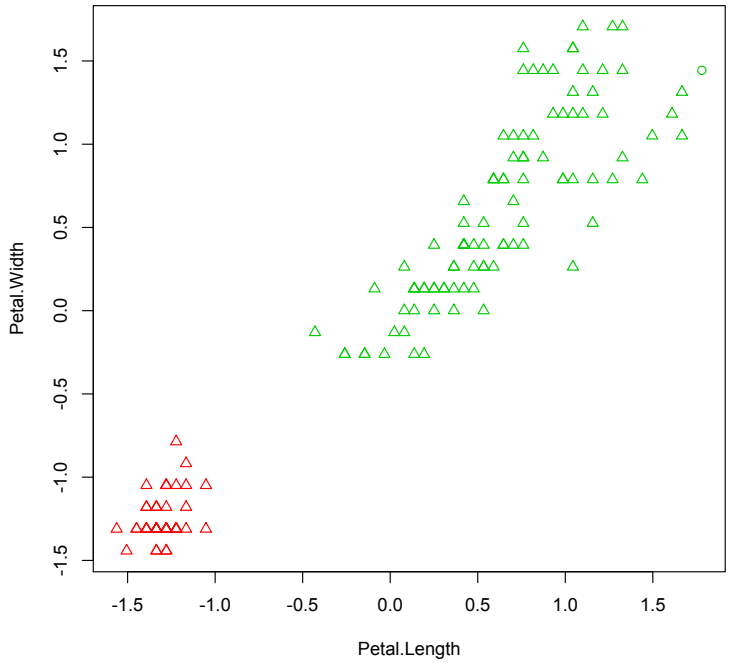
GGplot of K-means cluster, using K=3 shows how the data points can be clustered together into 3 clusters by color.



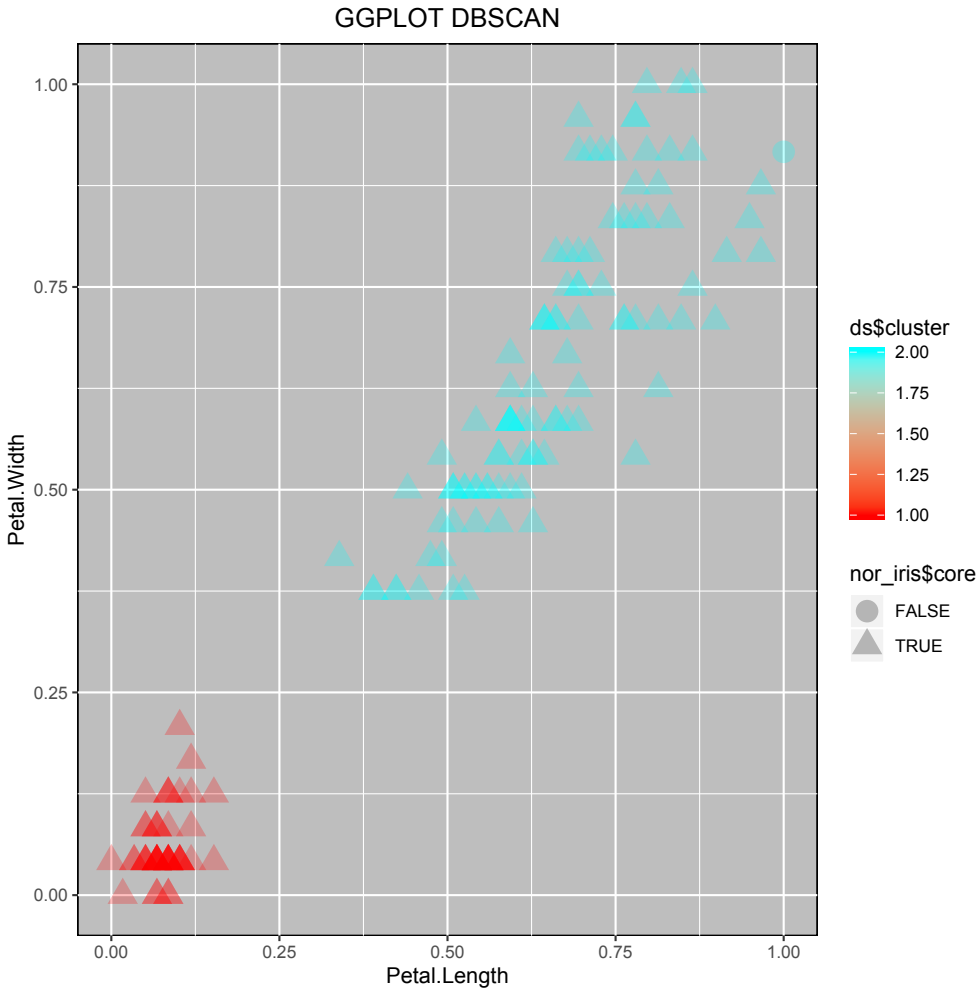
GGPlot showing end result of animation for Kmeans clusters with K=3 shows how the data points cluster into three distinct groups. However, the second and third groups are close together.



This plot from DBScan results shows that when using the Density-Based clustering algorithms, it favors 2 cluster groups.



This GGPlot of the DBScan results shows again that when using the Density-Based clustering algorithms, it favors 2 cluster groups.

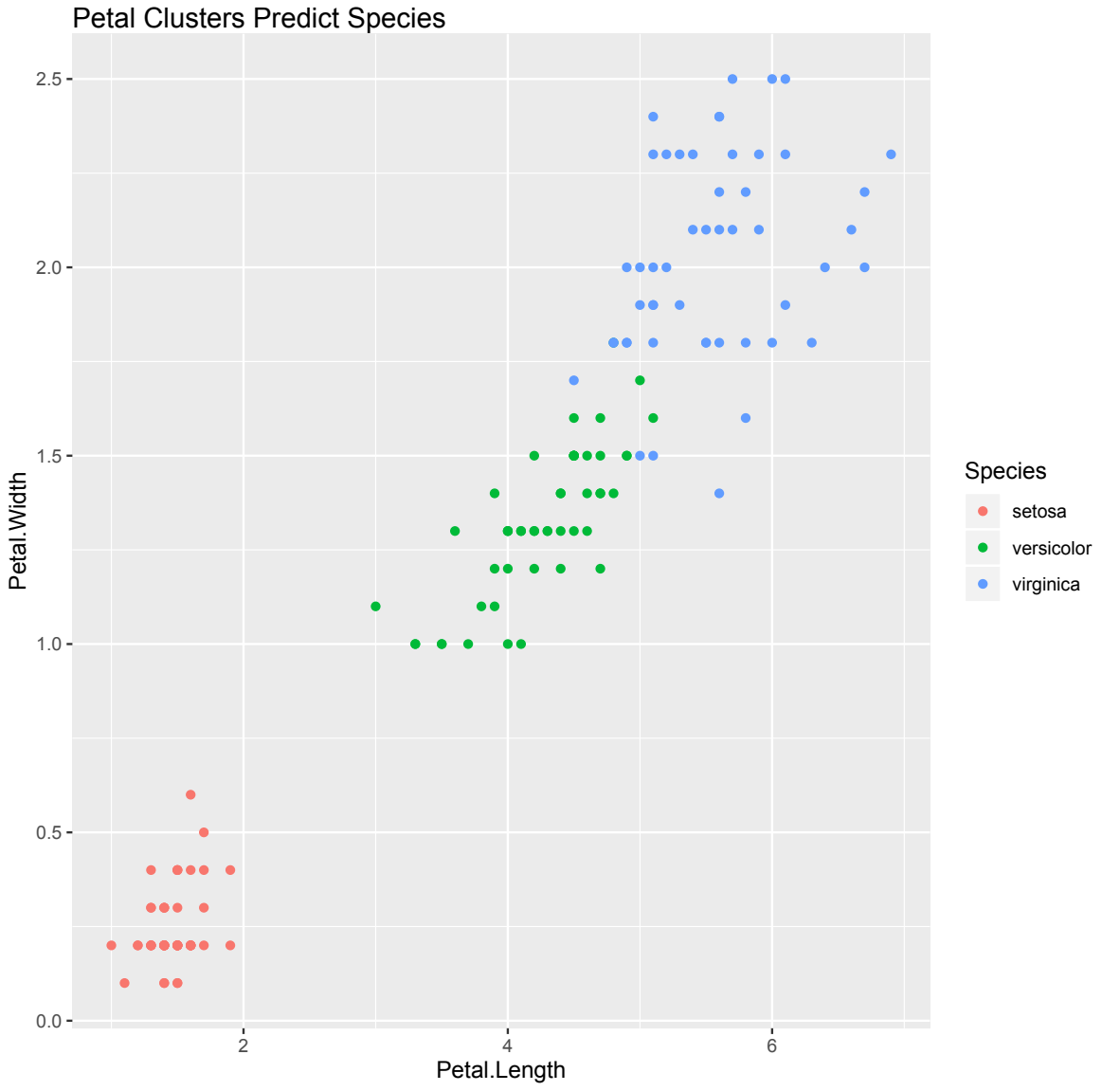


4. The K-means algorithm will have the best results when data has clusters that are not nested and can be exclusively partitioned into spherical or eliptical shapes. If the data does not cluster easily into circular shapes, K-means may not perform as optimal as other clustering methods. Also, if data is not normalized, it can give skewed results. Results can also be different depending on the starting point chosen for analysis, so solutions may vary when run repetitively. There is no guarantee for a global optimum using K-means.

To use the K-means algorithm, one must first pre-define how many clusters to begin analysis with. Different numbers of clusters will give different results. In order to determine the best possible number of clusters, there are a few methods that can be

executed. One method is to try different numbers and see what gets the best results. Generally, a lower total within-cluster sum of squares (aka sum of squares error) will give a better result, however there is a cut-off point that needs to be determined. Plotting an elbow curve showing the total sum of squared error vs the number of clusters can visually show a good cut-off point at the elbow, which will be a good number of clusters. One can also use the density-based method to see where data points are most densely clustered together. The K-means algorithm is a good classic method that is commonly used and gives good results and usable patterns.

5. Clustering can give insight into patterns within the data. For example, in this lab we used the iris dataset and can see that there are 2-3 clusters. If we dig deeper into these clusters we can see that a data point's cluster membership can predict its flower species. If we add species as a color dimension to the ggplot scatter plot, you can see that the species are clustered together by petal size. So,



one could predict the species based upon the size of the petal. However, there are some close numbers between virginica and versicolor species where the prediction could possibly be wrong, making it a prediction and not an absolute look-up table.