

Lab 4

Beatrice Dang

2024-09-27

```
library(ggplot2)
abalone <- read.csv(url("https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"))
attach(abalone)

colnames(abalone) <- c("sex", "length", 'diameter', 'height', 'whole_weight', 'shucked_wieght', 'viscera_weight', 'shell_weight', 'rings')
abalone$age.group <- cut(abalone$rings, br=c(0,8,11,35), labels = c("young", 'adult', 'old'))
abalone.norm <- abalone[, -1]
View(abalone)
```

EXERCISE 1: Naive Bayes on Abalone Dataset

```
library("e1071")

classifier<-naiveBayes(abalone[,5:8], abalone[,10])

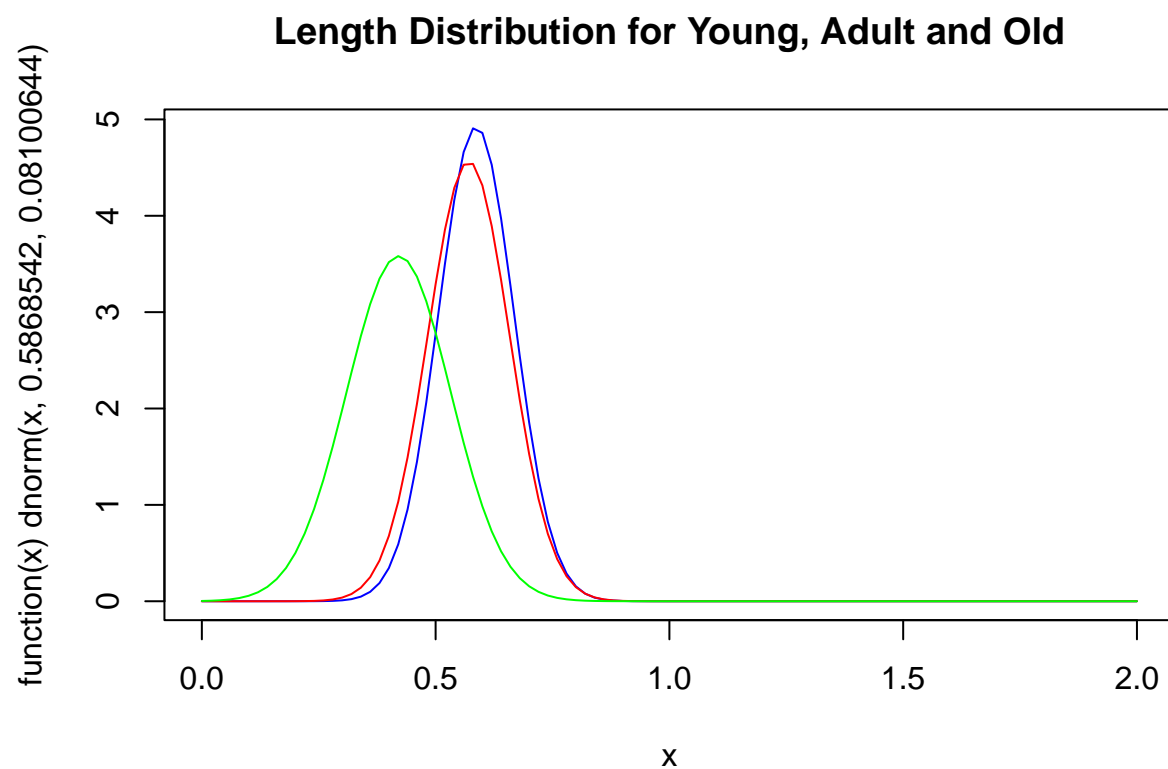
table(predict(classifier, abalone[,5:8]), abalone[,10], dnn=list('predicted','actual'))
```

```
##           actual
## predicted young adult  old
##    young  1164   507  192
##    adult   238  1021  489
##    old         5   282  279
```

```
classifier$tables$length
```

```
## NULL
```

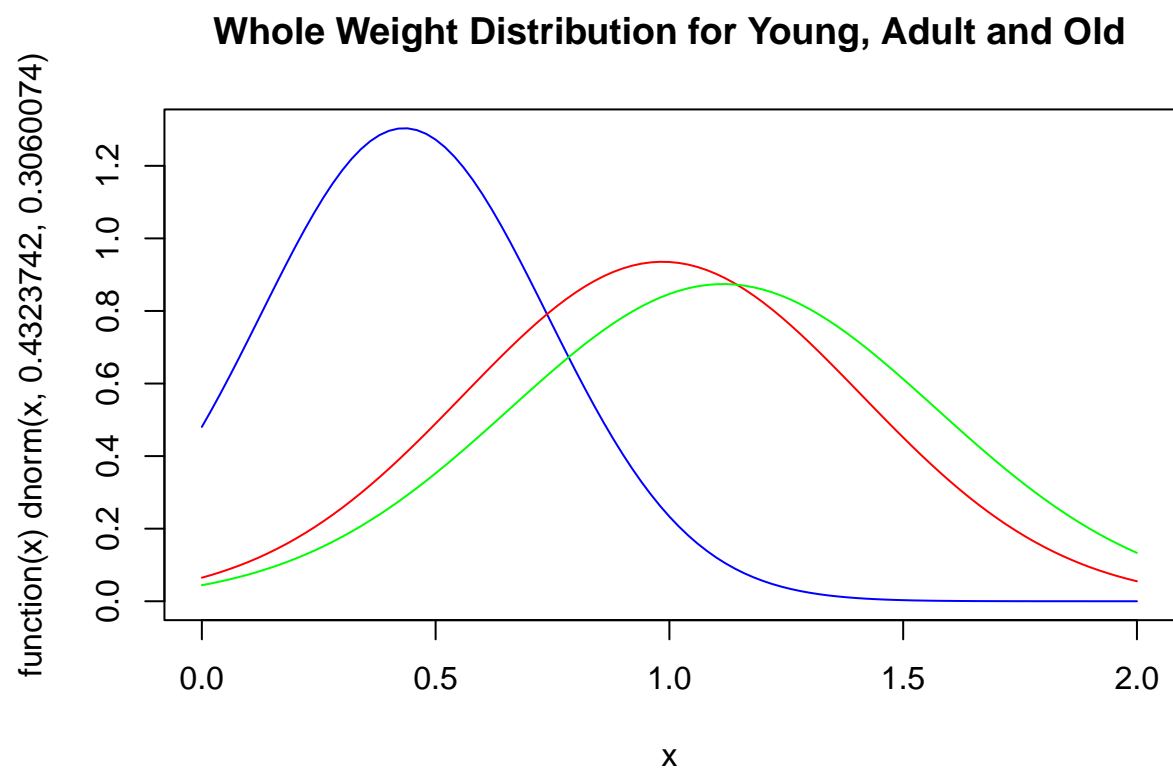
```
plot(function(x) dnorm(x, .5868542, .08100644), 0, 2, col="blue", main="Length Distribution for Young, Adult, Old")
curve(dnorm(x, .5707182, .08740980), add=TRUE, col="red")
curve(dnorm(x, .4209915, .11137474), add=TRUE, col="green")
```



```
classifier$tables$whole_weight
```

```
##           whole_weight
## abalone[, 10]      [,1]      [,2]
##      young 0.4323742 0.3060074
##      adult 0.9850878 0.4264315
##      old   1.1148922 0.4563715
```

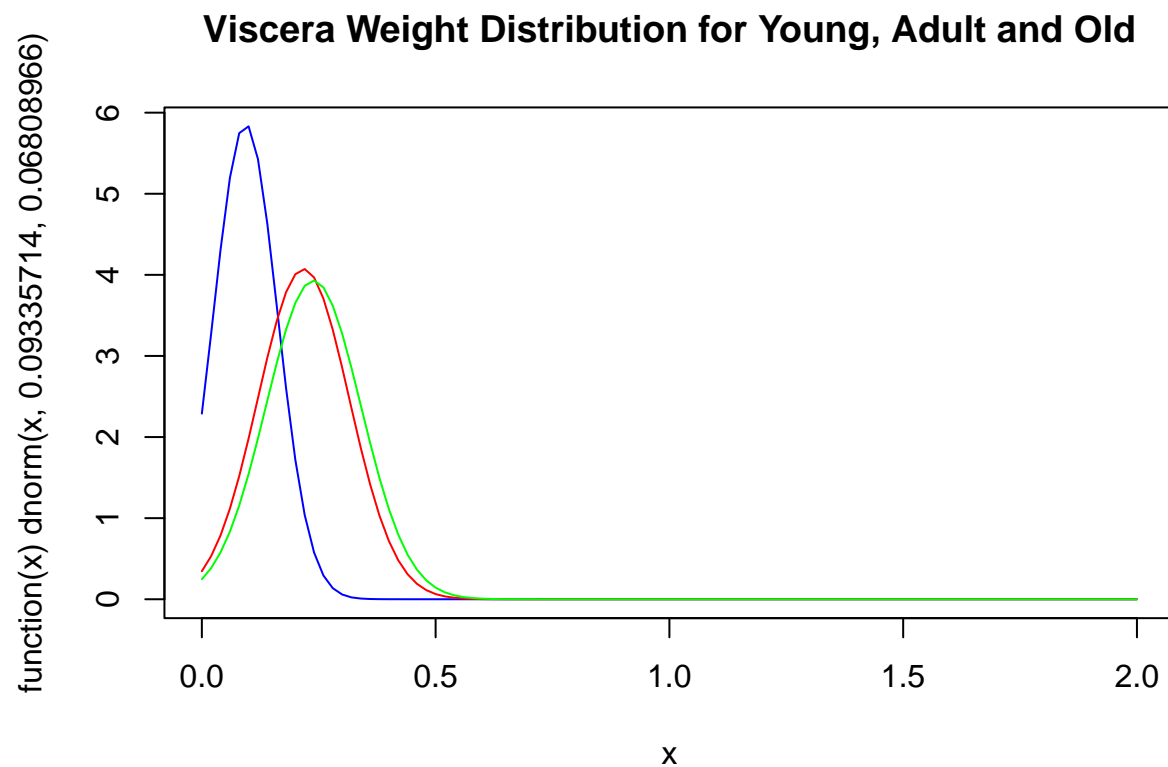
```
plot(function(x) dnorm(x, .4323742, .3060074), 0, 2, col="blue", main="Whole Weight Distribution for Young")
curve(dnorm(x, .9850878, .4264315), add=TRUE, col="red")
curve(dnorm(x, 1.1148922, .4563715), add=TRUE, col="green")
```



```
classifier$tables$viscera_wieght
```

```
##          viscera_wieght
## abalone[, 10]      [,1]      [,2]
##      young 0.09335714 0.06808966
##      adult 0.21755497 0.09793969
##      old   0.23876198 0.10148315
```

```
plot(function(x) dnorm(x, .09335714, .06808966), 0, 2, col="blue", main="Viscera Weight Distribution for Abalone")
curve(dnorm(x, .21755497, .09793969), add=TRUE, col="red")
curve(dnorm(x, .23876198, .10148315), add=TRUE, col="green")
```



K-nearest Neighbors

```
library(class)
iris <- read.csv("C:\\Users\\bmd\\Downloads\\iris.csv", header=TRUE)

attach(iris)

s_iris <- sample(150,105)

iris.train <-iris[s_iris,]
iris.test <- iris[-s_iris,]

sqrt(150)
```

```
## [1] 12.24745
```

```
k=13

KNNpred <- knn(train = iris.train[2:3], test = iris.test[2:3], cl = iris.train$Species, k = k)

contingency.table <- table(KNNpred,iris.test$Species)

contingency.matrix = as.matrix(contingency.table)

sum(diag(contingency.matrix))/length(iris.test$Species)
```

```
## [1] 0.7333333
```

```
accuracy <- c()
ks <- c(3,7,8,10)

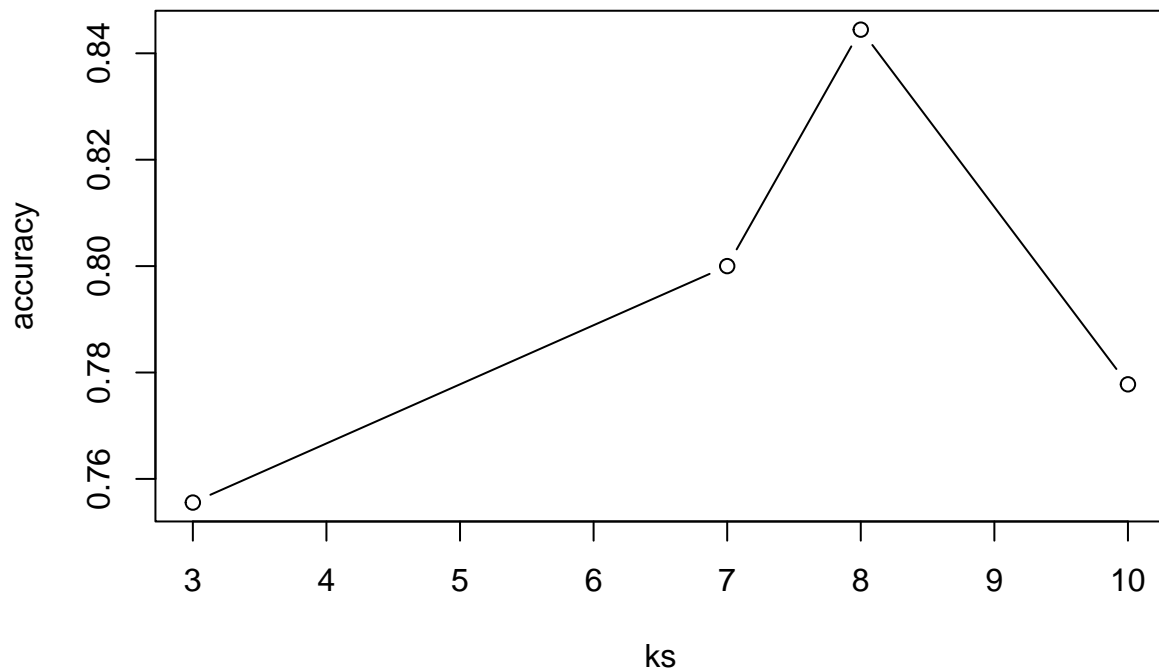
for (k in ks) {

  KNNpred <- knn(train = iris.train[2:3], test = iris.test[2:3], cl = iris.train$Species, k = k)
  cm = as.matrix(table(Actual=KNNpred, Predicted = iris.test$Species, dnn=list('predicted','actual')))

  accuracy <- c(accuracy,sum(diag(cm))/length(iris.test$Species))

}

plot(ks,accuracy,type = "b")
```



```
KNNpred <- knn(train = iris.train[3:4], test = iris.test[3:4], cl = iris.train$Species, k = k)
contingency.table <- table(KNNpred,iris.test$Species)
contingency.matrix = as.matrix(contingency.table)
sum(diag(contingency.matrix))/length(iris.test$Species)
```

```
## [1] 0.9777778
```

```

accuracy <- c()
ks <- c(3,7,8,10)

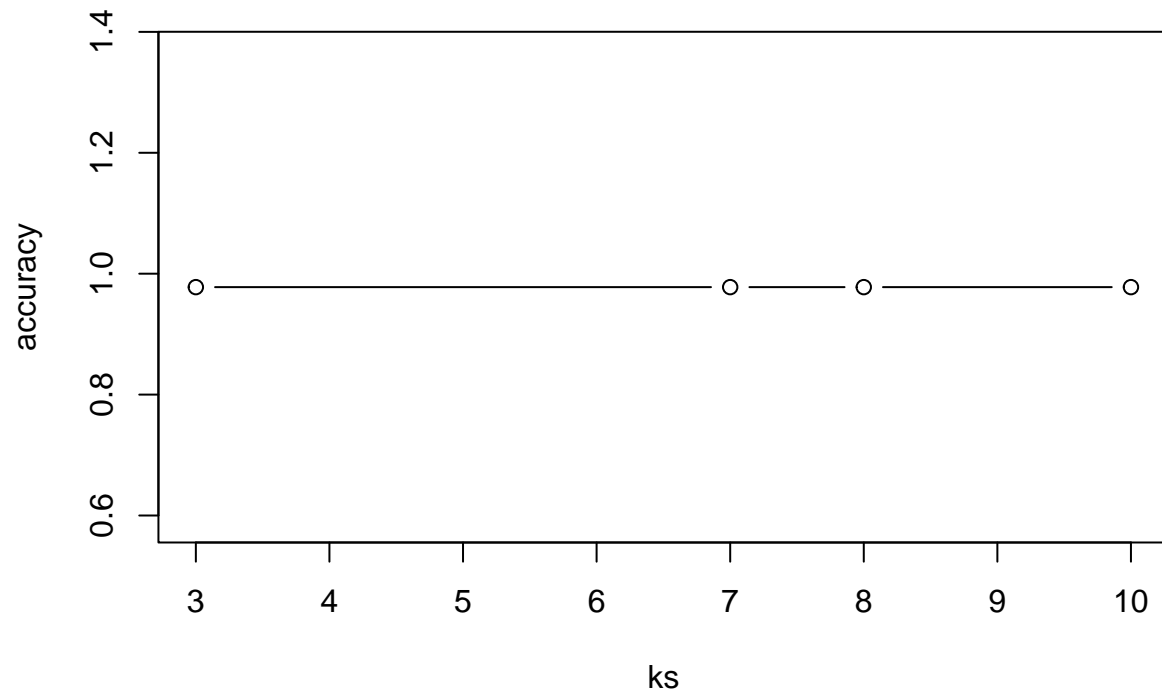
for (k in ks) {

  KNNpred <- knn(train = iris.train[3:4], test = iris.test[3:4], cl = iris.train$Species, k = k)
  cm = as.matrix(table(Actual=KNNpred, Predicted = iris.test$Species, dnn=list('predicted','actual')))

  accuracy <- c(accuracy,sum(diag(cm))/length(iris.test$Species))
}

plot(ks,accuracy,type = "b")

```

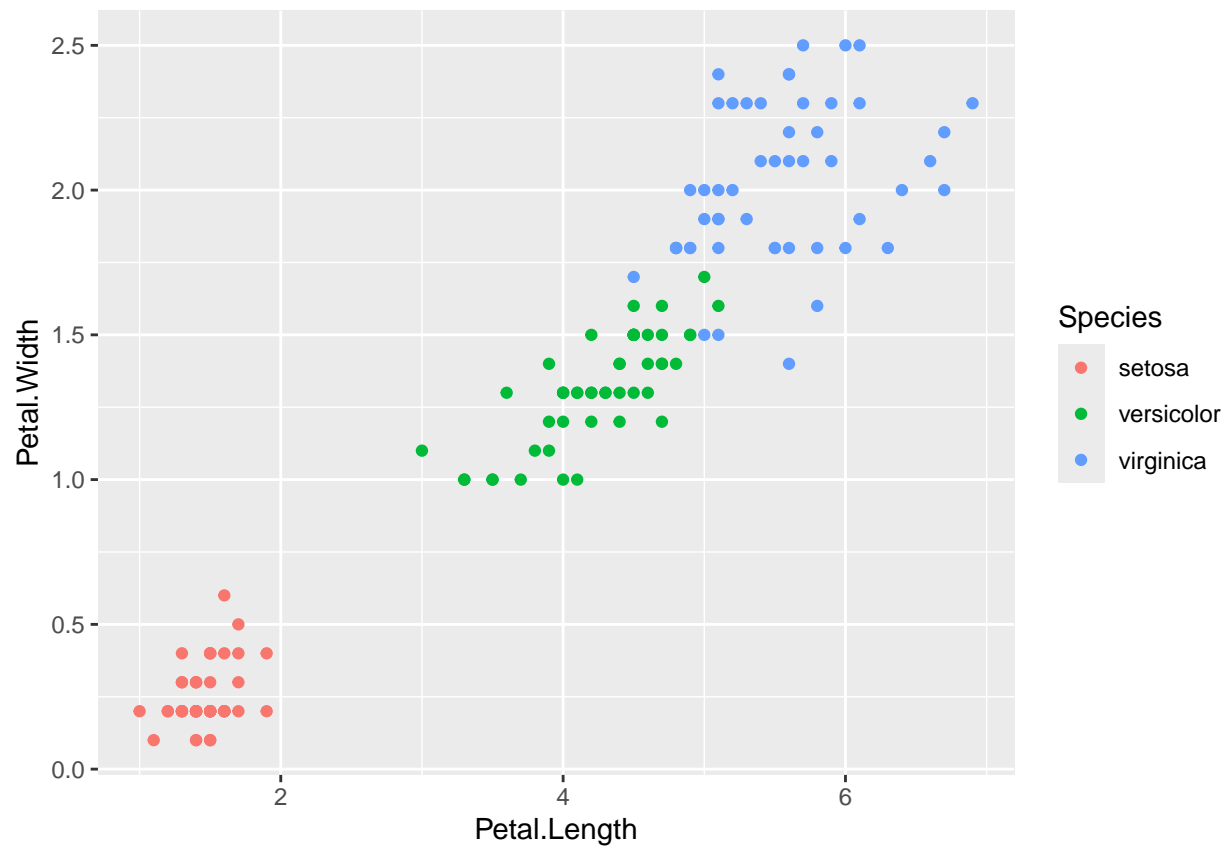


K-Means

```

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, colour = Species)) + geom_point()

```

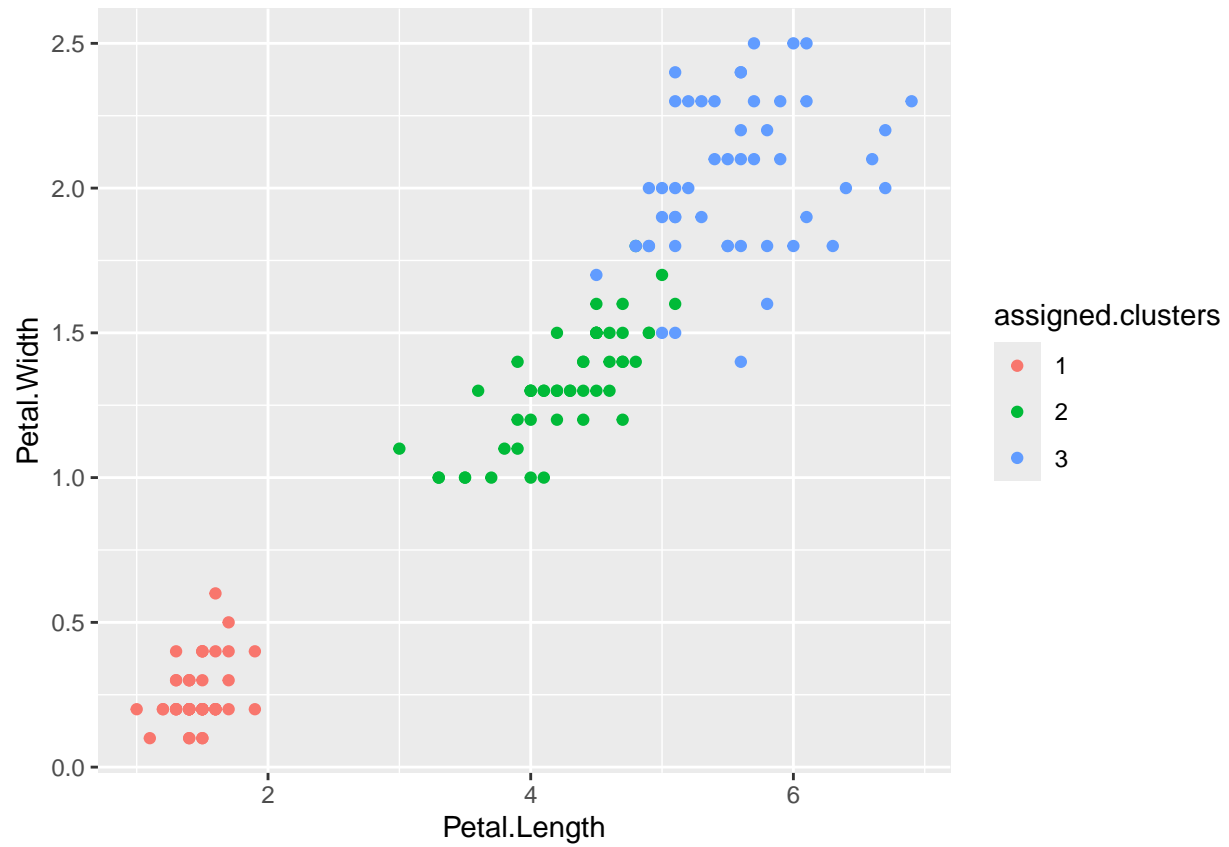


```
set.seed(123)

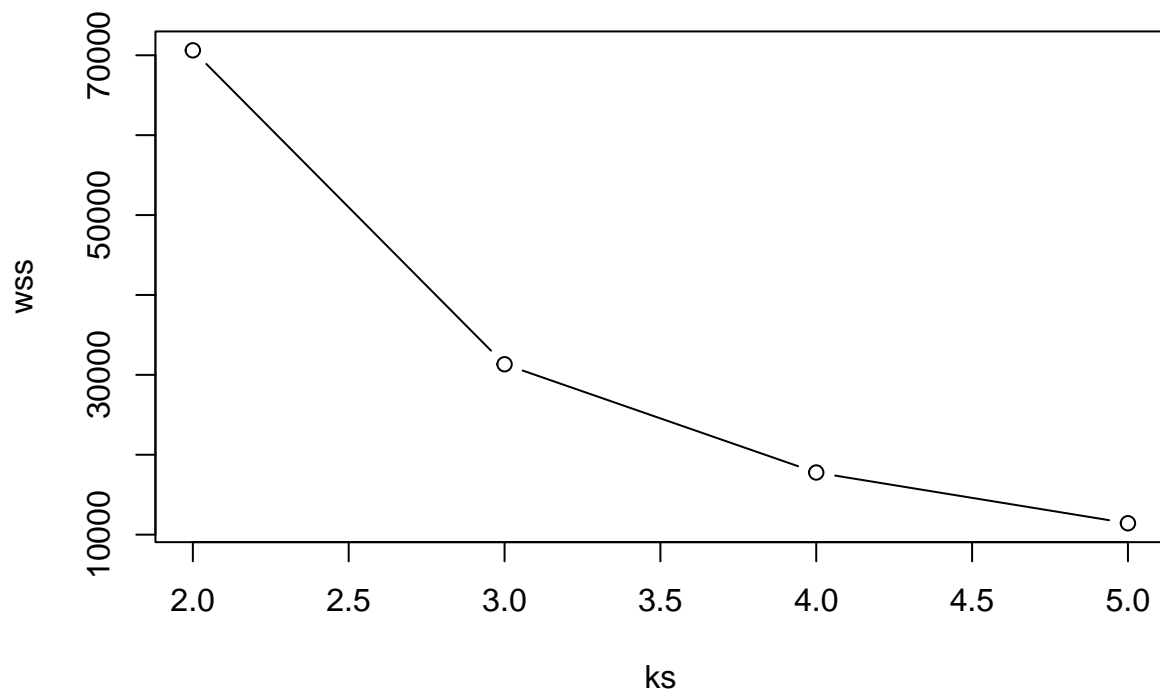
iris.km <- kmeans(iris[, -6], centers = 3)

assigned.clusters <- as.factor(iris.km$cluster)

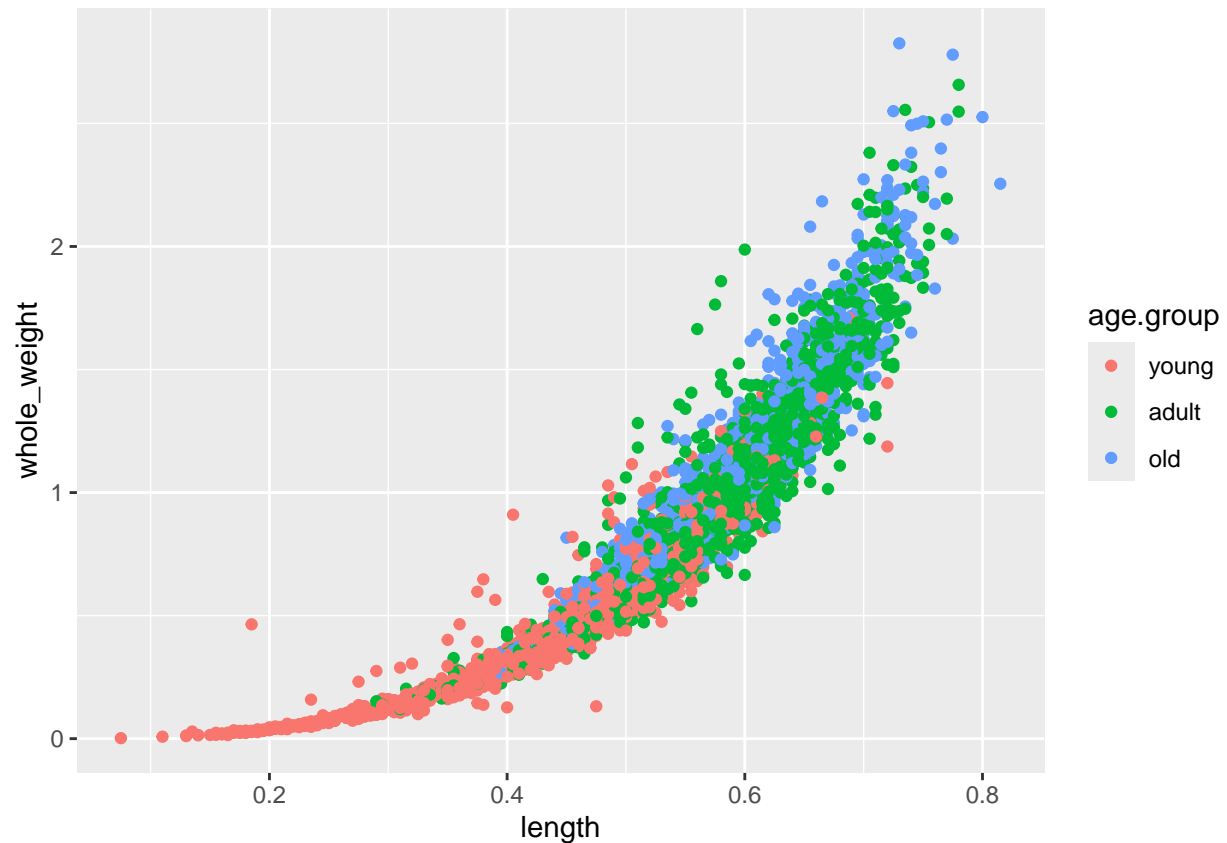
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, colour = assigned.clusters)) + geom_point()
```



```
wss <- c()
ks <- c(2,3,4,5)
for (k in ks) {
  iris.km <- kmeans(iris[, -6], centers = k)
  wss <- c(wss, iris.km$tot.withinss)
}
plot(ks, wss, type = "b")
```

```
ggplot(abalone, aes(x = length, y = whole_weight, colour = age.group)) +  
  geom_point()
```



```
## set random number generator start value
set.seed(123)

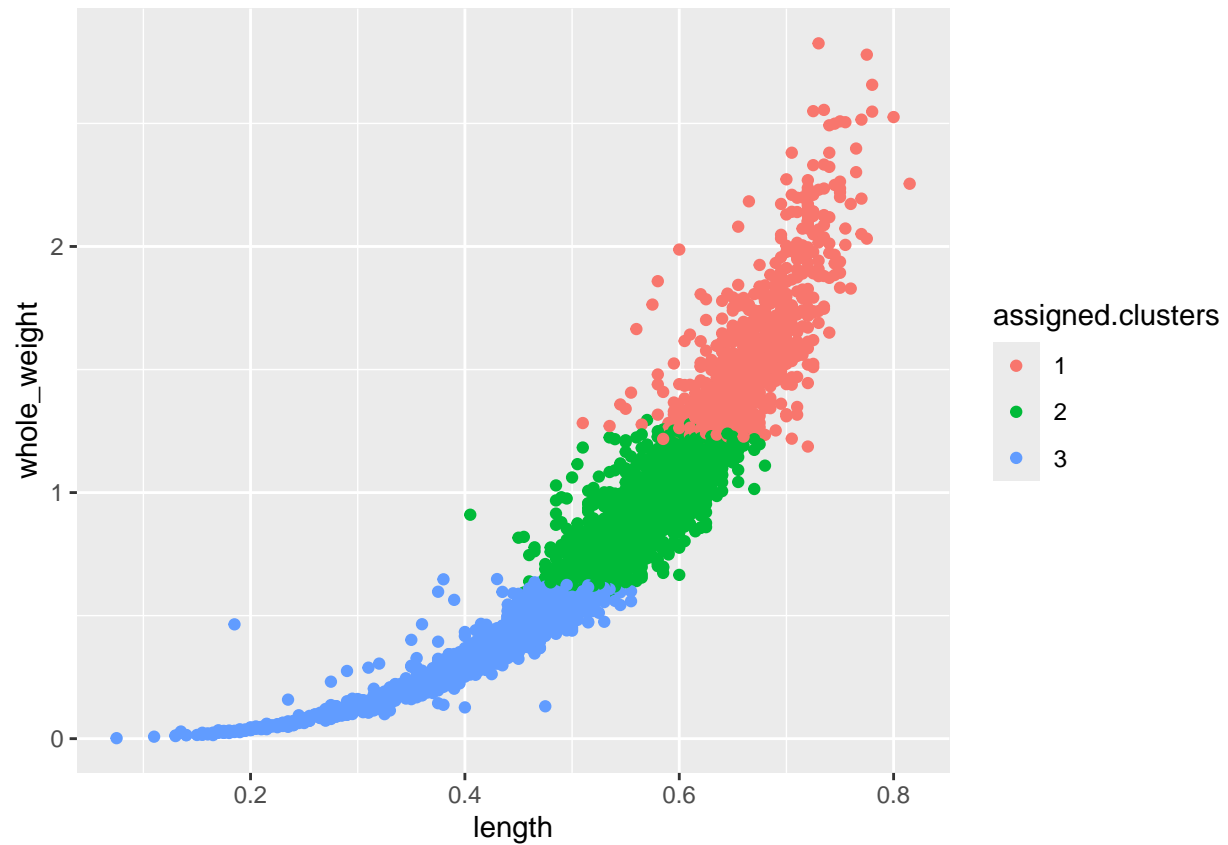
## train kmeans
abalone.km <- kmeans(abalone[2:8], centers = 3)

## WCSS: total within cluster sum of squares
abalone.km$tot.withinss
```

```
## [1] 271.3735
```

```
## get and plot clustering output
assigned.clusters <- as.factor(abalone.km$cluster)

ggplot(abalone, aes(x = length, y = whole_weight, colour = assigned.clusters)) +
  geom_point()
```



```
wcss1 <- c()
ks1 <- c(2,3,4,5,6,7,8)

for (k in ks) {

  abalone.km <- kmeans(abalone[2:8], centers = k)

  wcss <- c(wss,abalone.km$tot.withinss)

}

plot(ks1,wcss1,type = "b")
```

