Project 2

Marcus McKenzie

2.1

Using the same data as in Question 2.2, use the ksvm or kknn function to find a good classifier: 1. using cross-validation (do this for the k-nearest-neighbors model; SVM is optional); and 2. splitting the data into training, validation, and test data sets (pick either KNN SVM; the other is optional).

#Part A

Load the knn library:

```
library(kknn)
```

Load and organize the data:

```
data <- read.table("credit_card_data.txt", header=FALSE, stringsAsFactors = FALSE)

set.seed(1)

rows <- sample(1:nrow(data),as.integer(0.7*nrow(data)))

training <- data[rows,]

testing <- data[-rows,]
```

Initialize Training data:

```
train.kknn(as.factor(V11)~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10, data = training, kmax = 50, scale = TRUE)
```

```
##
## Call:
## train.kknn(formula = as.factor(V11) ~ V1 + V2 + V3 + V4 + V5 +     V6 + V7 + V8 + V9 + V10, data = t
##
## Type of response variable: nominal
## Minimal misclassification: 0.1509847
## Best kernel: optimal
## Best k: 12
```

```
pred_training <- rep(0, (nrow(training)))
acc_training <- 0
```

Create model from training data:

```
for (i in 1:nrow(training)){
  model=kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10,training,training[i,],k=12,kernel="optimal", scale = TRU

  pred_training[i] <- as.integer(fitted(model)+.5)}

acc_training <- sum(pred_training == training[,11]) / nrow(training)
```

Predictions on testing data:

```
pred_testing <- rep(0,(nrow(testing)))
acc_testing <- 0

for (i in 1:nrow(testing)){
  model=kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10,testing[-i,],testing[i,],k=12,kernel="optimal", scale =

  pred_testing[i] <- as.integer(fitted(model)+.5)

}

acc_testing <- sum(pred_testing == testing[,11]) / nrow(testing)
```

Results:

```
acc_training
```

```
## [1] 0.9190372
```

```
acc_testing
```

```
## [1] 0.8020305
```

Part B

Load knn library:

```
rm(list = ls())

library(kknn)
```

Load and Organize data:

```
data <- read.table("credit_card_data.txt", header=FALSE, stringsAsFactors = FALSE)

set.seed(1)

rows <- sample(1:nrow(data),as.integer(0.8*nrow(data)))

training <- data[rows,]
```

```
rem <- data[-rows,]

row2 <- sample(1:nrow(rem),as.integer(.5*nrow(rem)))

validation = rem[row2,]

testing = rem[-row2,]
```

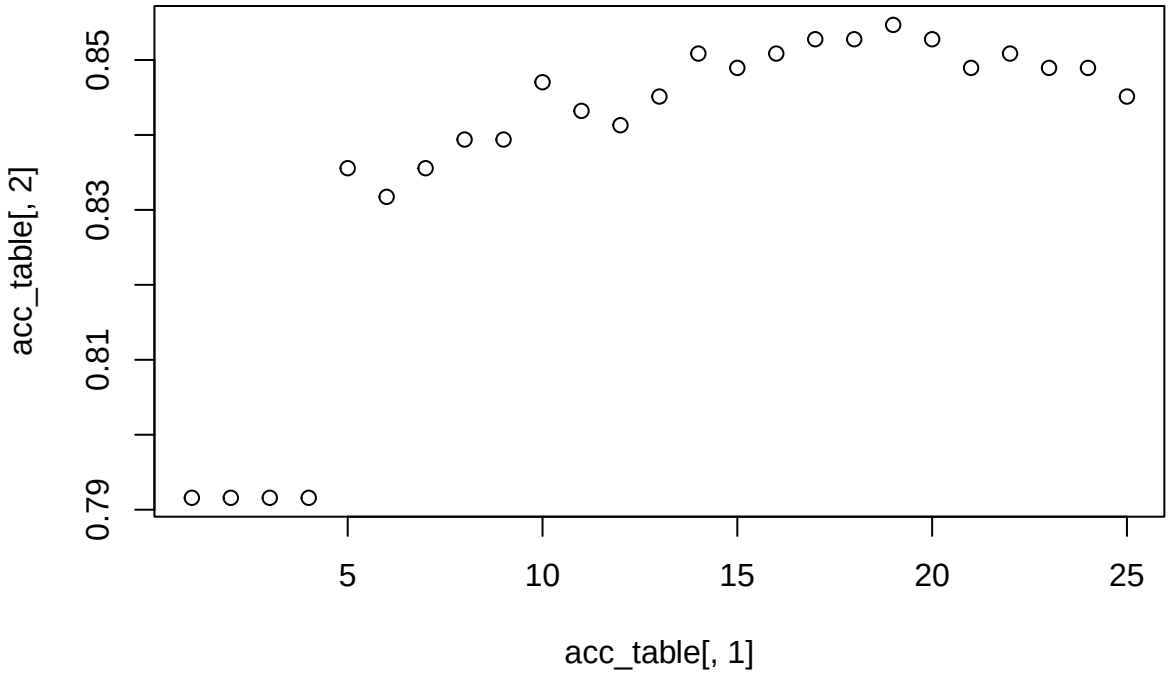
Training model with training data:

```
pred_training <- rep(0,(nrow(training)))
acc_training <- 0
X <- 0

acc_table <- data.frame(matrix(nrow = 25, ncol = 2))
colnames(acc_table) <- c("K", "Accuracy")

for (X in 1:25){

  for (i in 1:nrow(training)){

    model=kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10,training[-i,],training [i,],k=X,kernel="optimal", scal

    pred_training[i] <- as.integer(fitted(model)+.5)
  }

  acc_training <- sum(pred_training == training[,11]) / nrow(training)

  acc_table[X, 1] <- X
  acc_table[X, 2] <- acc_training
}

acc_table
```

```
##     K  Accuracy
## 1   1 0.7915870
## 2   2 0.7915870
## 3   3 0.7915870
## 4   4 0.7915870
## 5   5 0.8355641
## 6   6 0.8317400
## 7   7 0.8355641
## 8   8 0.8393881
## 9   9 0.8393881
## 10 10 0.8470363
## 11 11 0.8432122
## 12 12 0.8413002
## 13 13 0.8451243
## 14 14 0.8508604
## 15 15 0.8489484
## 16 16 0.8508604
## 17 17 0.8527725
## 18 18 0.8527725
```

```
## 19 19 0.8546845
## 20 20 0.8527725
## 21 21 0.8489484
## 22 22 0.8508604
## 23 23 0.8489484
## 24 24 0.8489484
## 25 25 0.8451243
```

```r
plot(acc_table[,1], acc_table[,2])
```



Validating model with validation data:

```r
pred_validation <- rep(0, nrow(validation))
acc_validation <- 0

x <- 0
acc_table_validation <-data.frame(matrix(nrow = 4, ncol = 2))

colnames(acc_table_validation) <- c("K", "Acc_Valid")

count <- 0

for (x in 12:15){

  count <- count + 1
```

```
  for (i in 1:nrow (validation)){

    model=kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10,validation[-i,],validation[i,],k=x,kernel="optimal", s
    pred_validation[i] <- as.integer(fitted(model)+.5)
  }


  acc_validation <- sum(pred_validation == validation[,11]) / nrow(validation)

  acc_table_validation[count, 1] <- x
  acc_table_validation[count, 2] <- acc_training

}

acc_table_validation
```

```
##    K Acc_Valid
## 1 12 0.8451243
## 2 13 0.8451243
## 3 14 0.8451243
## 4 15 0.8451243
```

Predictions using testing data:

```
pred_testing <- rep(0, (nrow(testing)))
acc_testing <- 0

for (i in 1:nrow(testing)){

  model=kknn(V11~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10,testing[-i,],testing[i,],k=12,kernel="optimal", scale =
  pred_testing[i] <- as.integer(fitted(model)+.5)

}

acc_testing <- sum(pred_testing == testing[,11]) / nrow(testing)
```

Results:

```
acc_training
```

```
## [1] 0.8451243
```

```
acc_testing
```

```
## [1] 0.8181818
```

2.1 Analysis: From the results we can see that the model performed fairly in both models. In the first test in which we used knn to determine the a classifier the results were fairly accurate but the results from the training set performed better than in the testing set. However, when a validation set was also implemented, this result reversed and the testing set actually performed significantly better than the testing set and than its previous classification.

2.2

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

Given how destructive earthquakes have been in California it could save lives to gather data about these earthquakes to determine which locations may be at the greatest risk of an earthquake.

Some predictors that could potentially be used are: the locations of epicenters, the magnitude, radius, depth, and cause of the earthquakes. It may be useful to look at the surrounding area for building designs, geology and population density to determine whether the effects of an earthquake will be damaging.

2.3

Use the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

Load libraries:

```
rm(list = ls())

library(kknn)
library(ggplot2)
```

Load and organize data:

```
data <- read.table("iris.txt", header=TRUE, stringsAsFactors = FALSE)

data
```

```
##     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
## 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
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
```

```
## 23          4.6         3.6          1.0         0.2     setosa
## 24          5.1         3.3          1.7         0.5     setosa
## 25          4.8         3.4          1.9         0.2     setosa
## 26          5.0         3.0          1.6         0.2     setosa
## 27          5.0         3.4          1.6         0.4     setosa
## 28          5.2         3.5          1.5         0.2     setosa
## 29          5.2         3.4          1.4         0.2     setosa
## 30          4.7         3.2          1.6         0.2     setosa
## 31          4.8         3.1          1.6         0.2     setosa
## 32          5.4         3.4          1.5         0.4     setosa
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
```

```
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 105         6.5         3.0          5.8         2.2  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 110         7.2         3.6          6.1         2.5  virginica
## 111         6.5         3.2          5.1         2.0  virginica
## 112         6.4         2.7          5.3         1.9  virginica
## 113         6.8         3.0          5.5         2.1  virginica
## 114         5.7         2.5          5.0         2.0  virginica
## 115         5.8         2.8          5.1         2.4  virginica
## 116         6.4         3.2          5.3         2.3  virginica
## 117         6.5         3.0          5.5         1.8  virginica
## 118         7.7         3.8          6.7         2.2  virginica
## 119         7.7         2.6          6.9         2.3  virginica
## 120         6.0         2.2          5.0         1.5  virginica
## 121         6.9         3.2          5.7         2.3  virginica
## 122         5.6         2.8          4.9         2.0  virginica
## 123         7.7         2.8          6.7         2.0  virginica
## 124         6.3         2.7          4.9         1.8  virginica
## 125         6.7         3.3          5.7         2.1  virginica
## 126         7.2         3.2          6.0         1.8  virginica
## 127         6.2         2.8          4.8         1.8  virginica
## 128         6.1         3.0          4.9         1.8  virginica
## 129         6.4         2.8          5.6         2.1  virginica
## 130         7.2         3.0          5.8         1.6  virginica
```

```
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

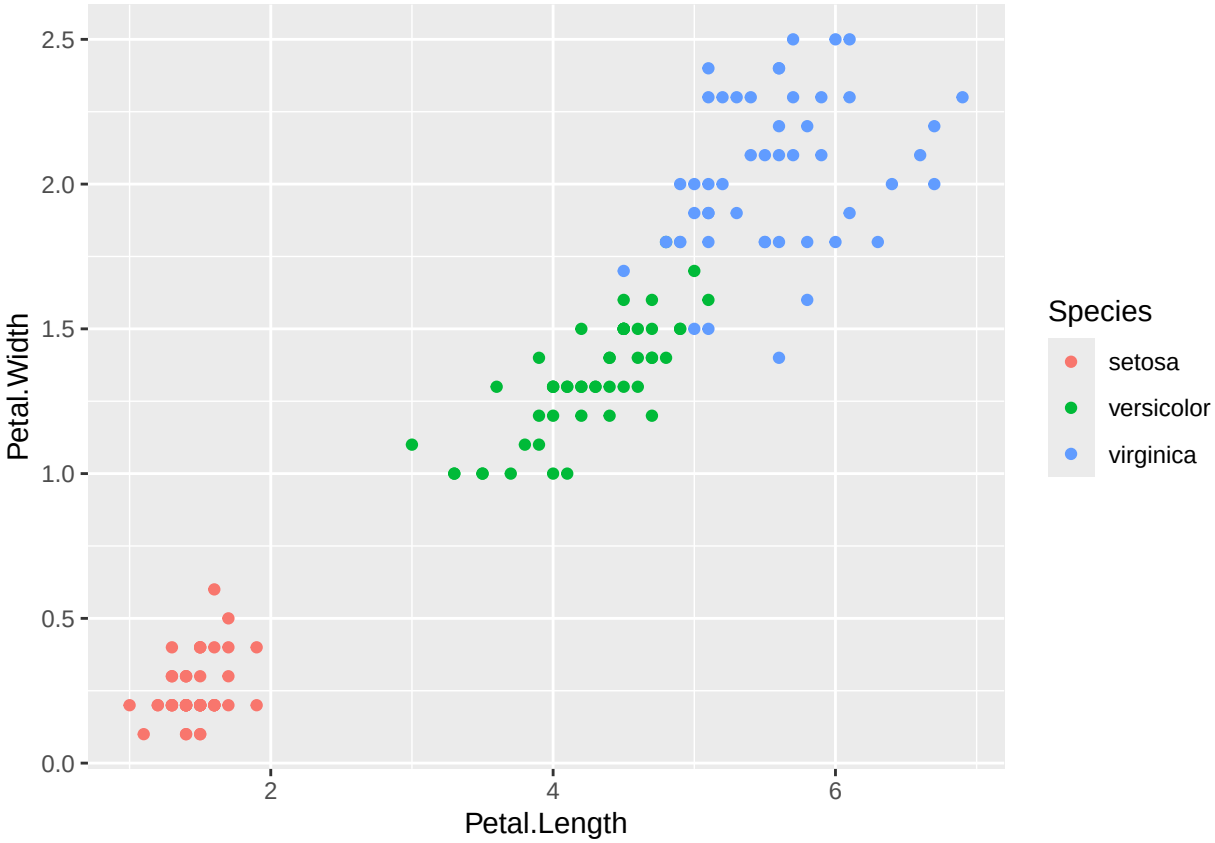
```
data <- data[,1:5]

head(data)
```

```
##   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
## 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
```

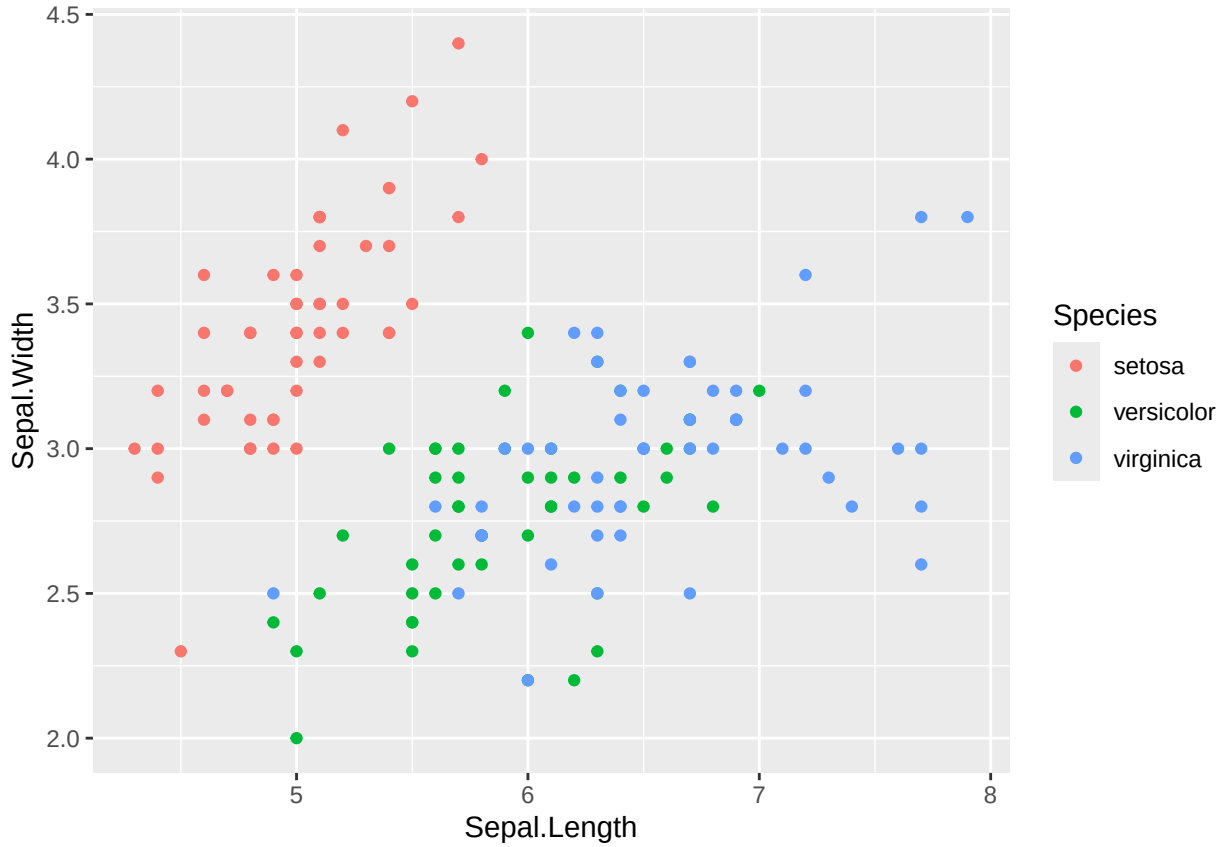
```
set.seed(1)
```

Analyze data:

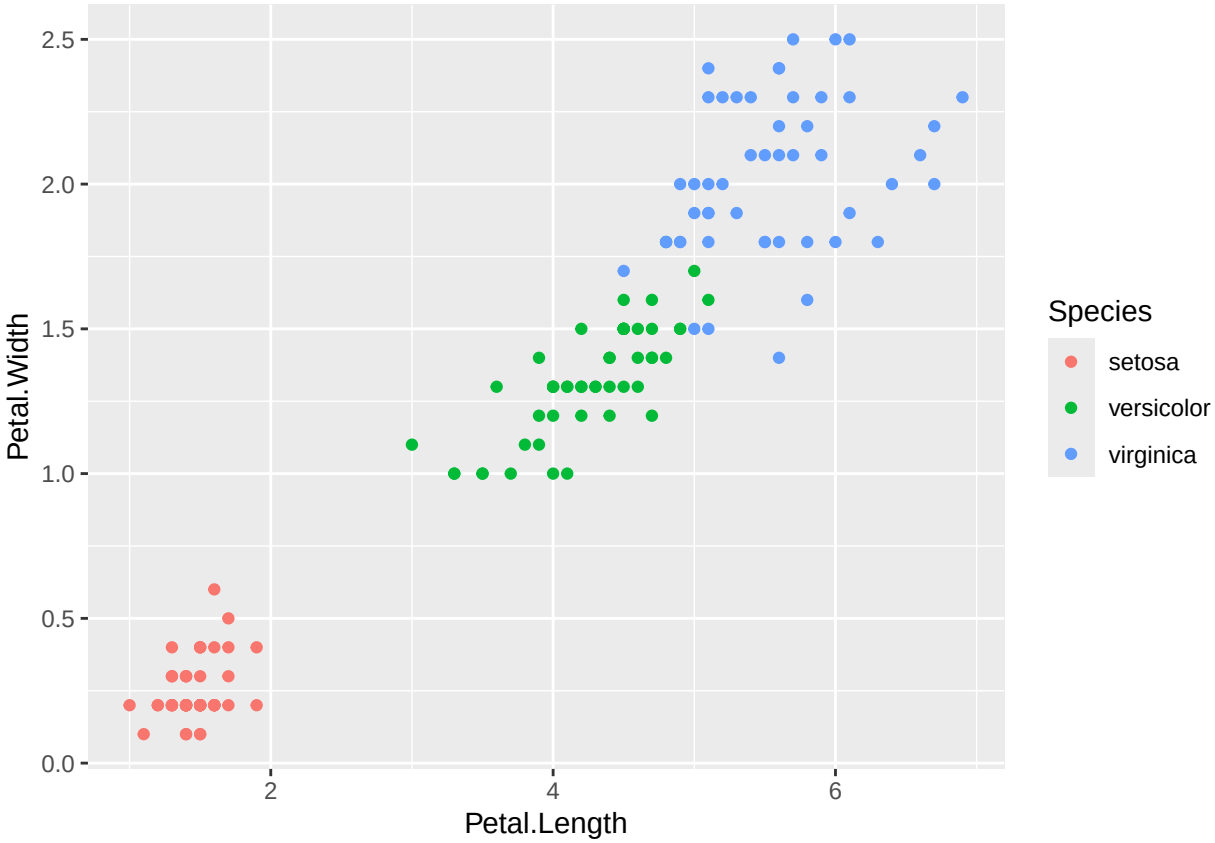
```
ggplot(data, aes(Petal.Length, Petal.Width, color= Species)) + geom_point()
```



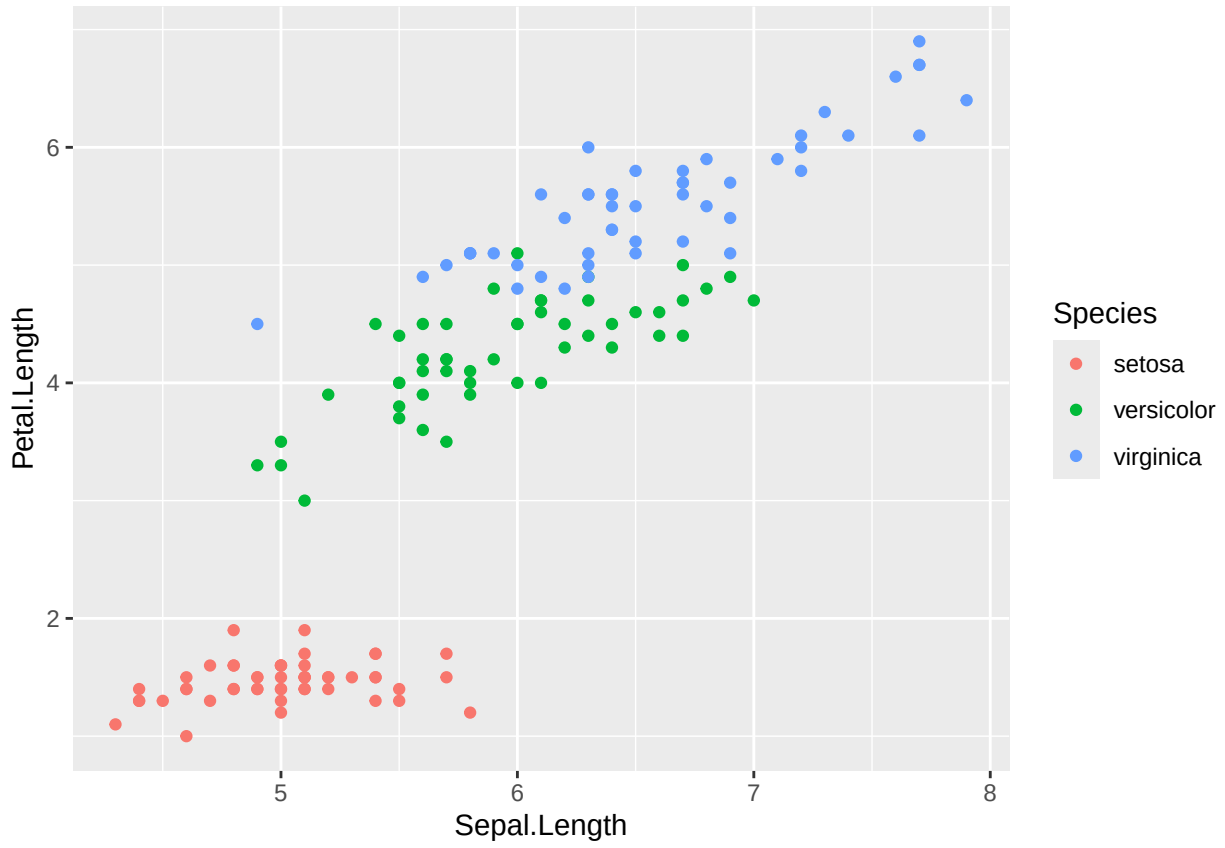

```
ggplot(data, aes(Sepal.Length, Sepal.Width, color= Species)) + geom_point()
```



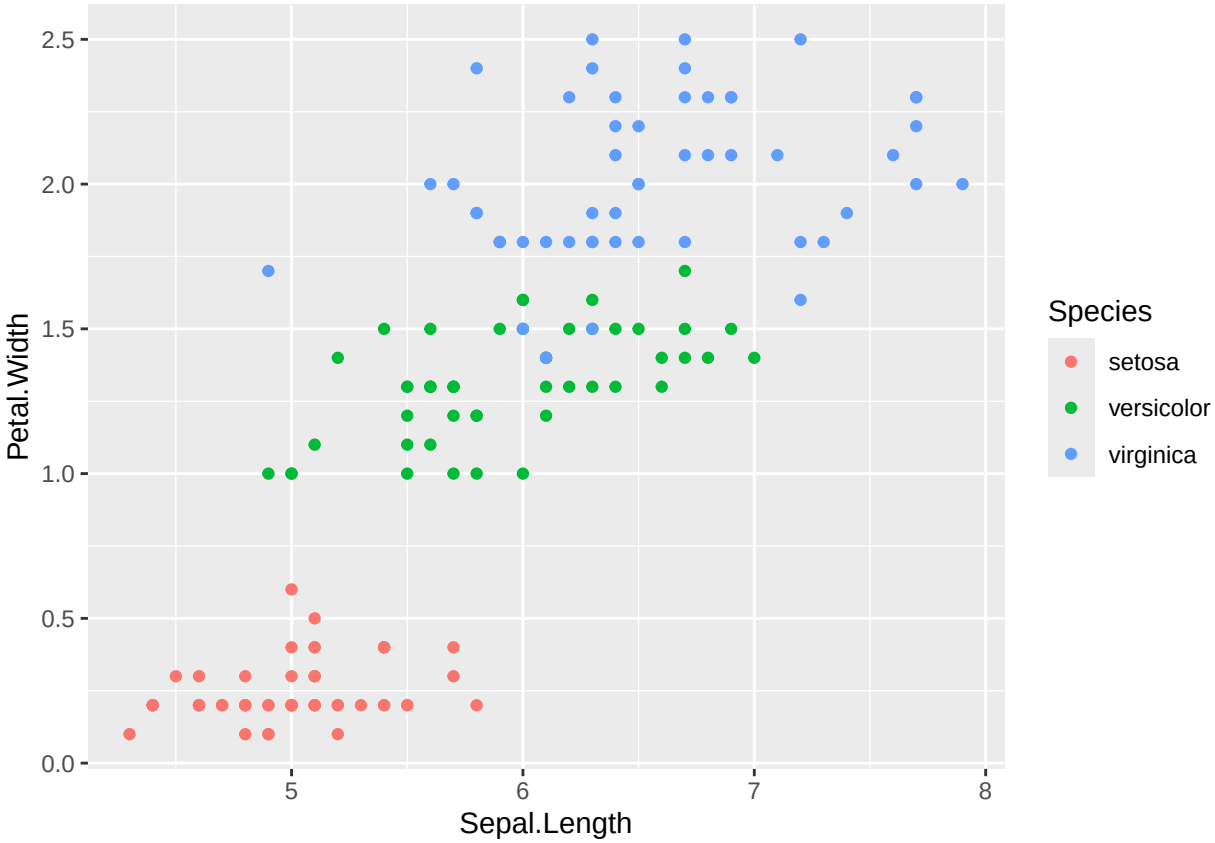

```
ggplot(data, aes(Petal.Length, Petal.Width, color= Species)) + geom_point()
```



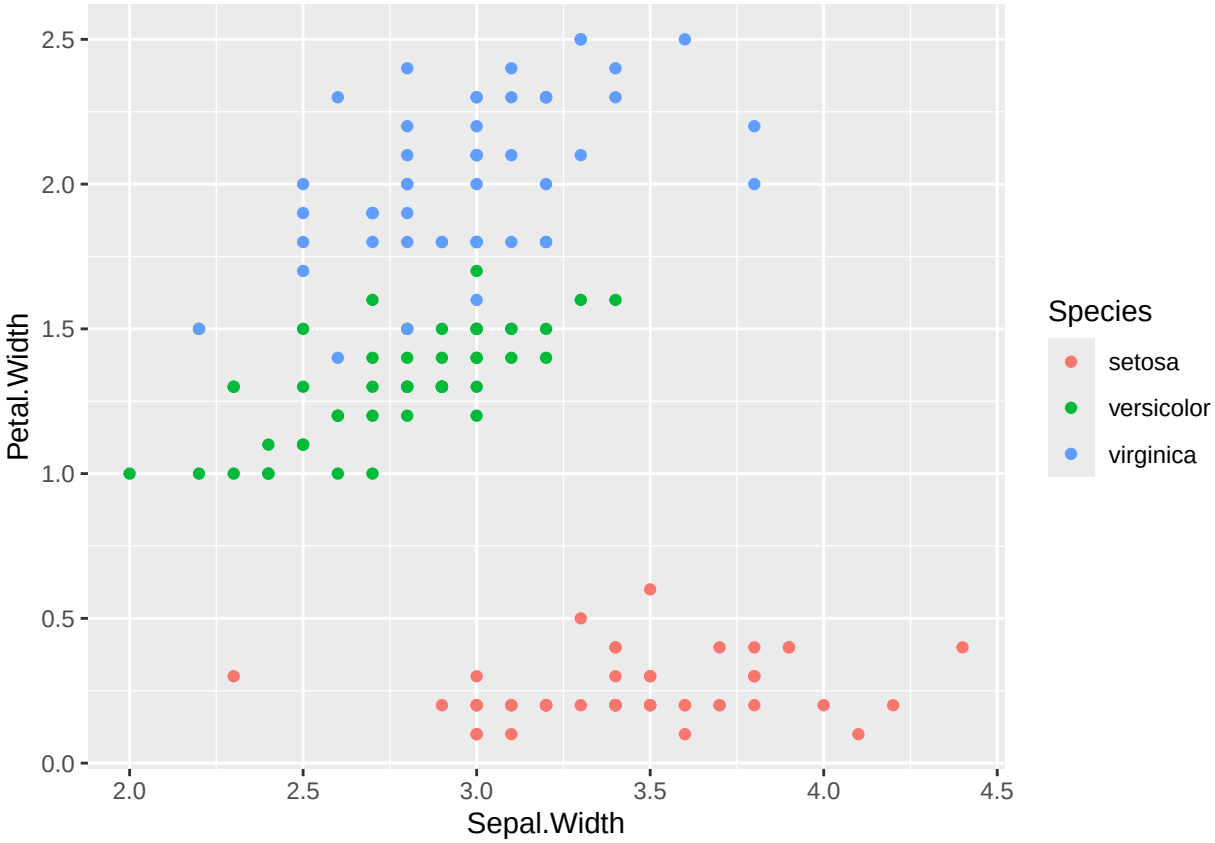

```
ggplot(data, aes(Sepal.Length, Petal.Length, color= Species)) + geom_point()
```



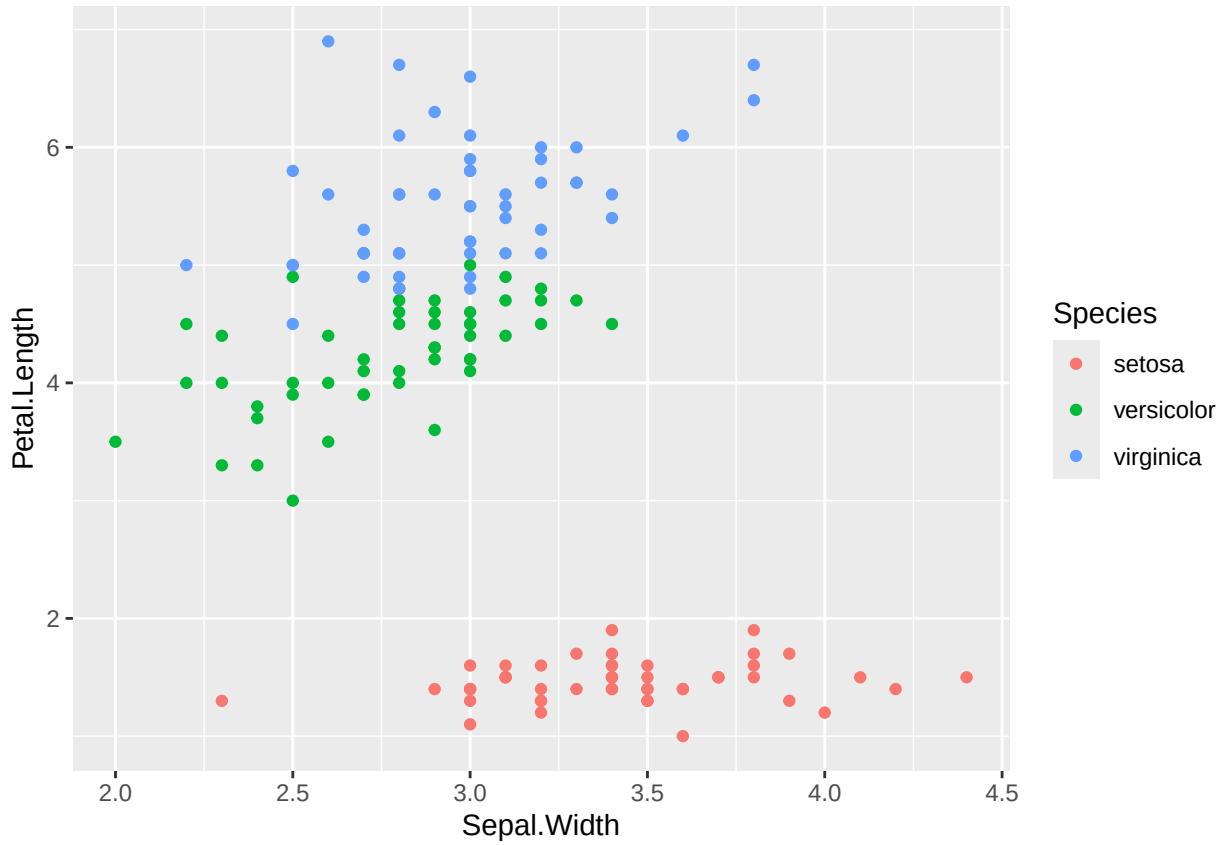

```
ggplot(data, aes(Sepal.Length, Petal.Width, color= Species)) + geom_point()
```




```
ggplot(data, aes(Sepal.Width, Petal.Width, color= Species)) + geom_point()
```




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



Creating model:

```
model<- kmeans(data[,3:4], 3)
```

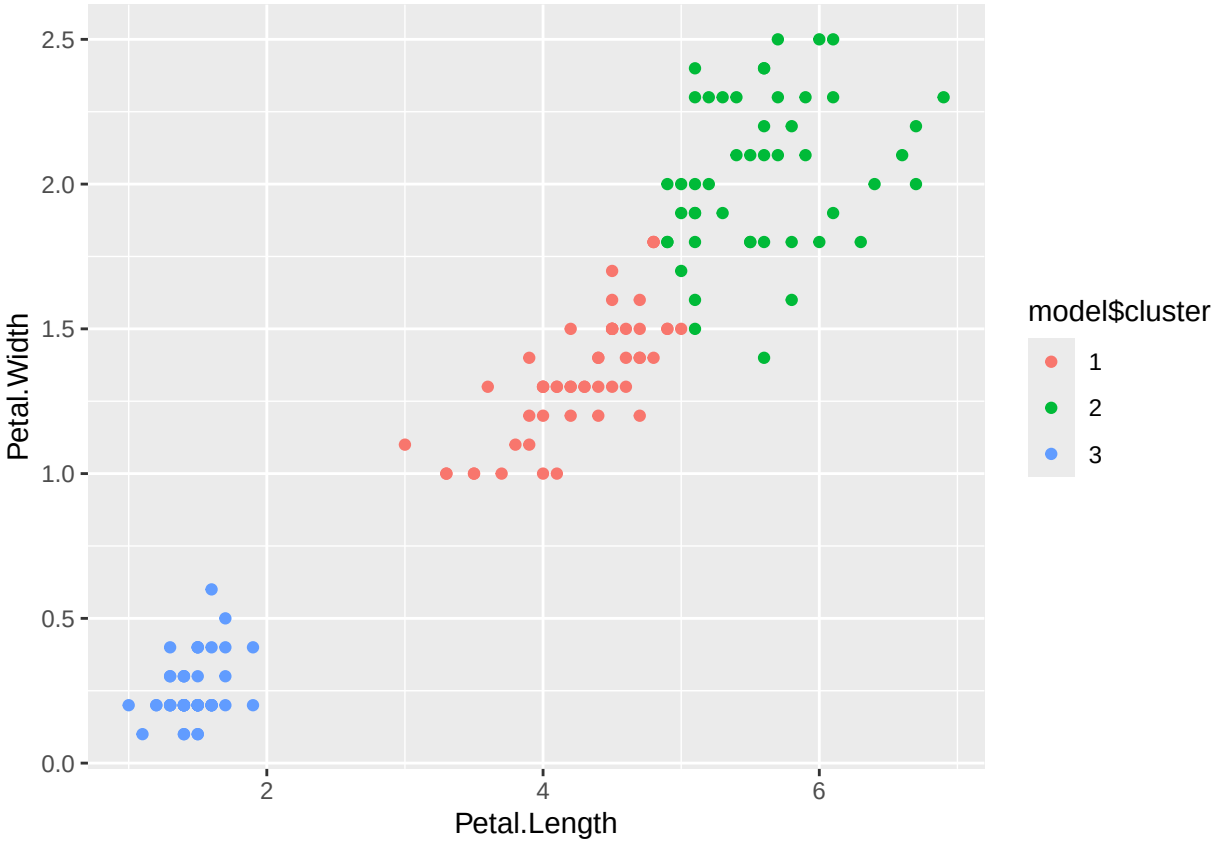
Scaled Data and Clusters:

```
sdata <- data
for (i in 1:4) { sdata[,i] <- (data[,i]-min(data[,i]))/(max(data[,i])-min(data[,i])) }


table(model$cluster, data$Species)
```

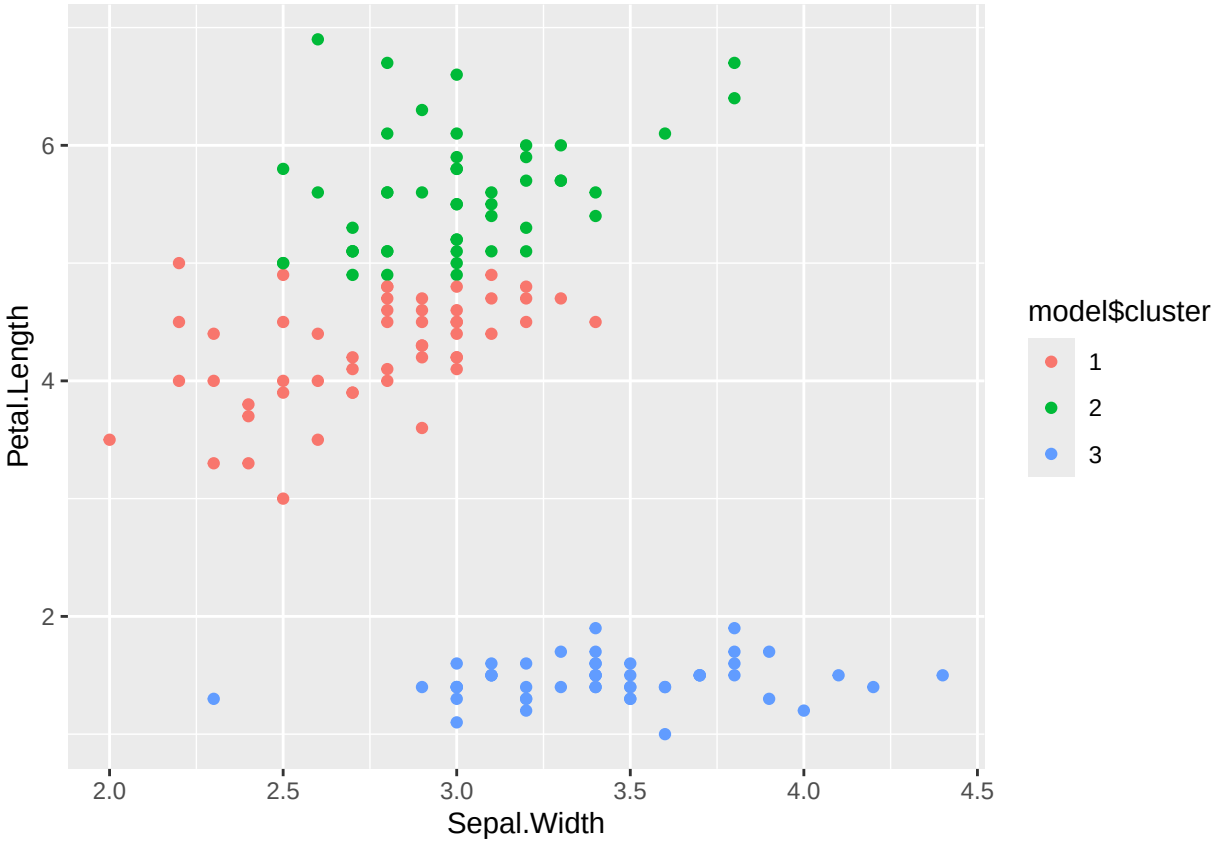
```
##
##     setosa versicolor virginica
##   1      0         48         4
##   2      0          2        46
##   3     50          0         0
```

Results:

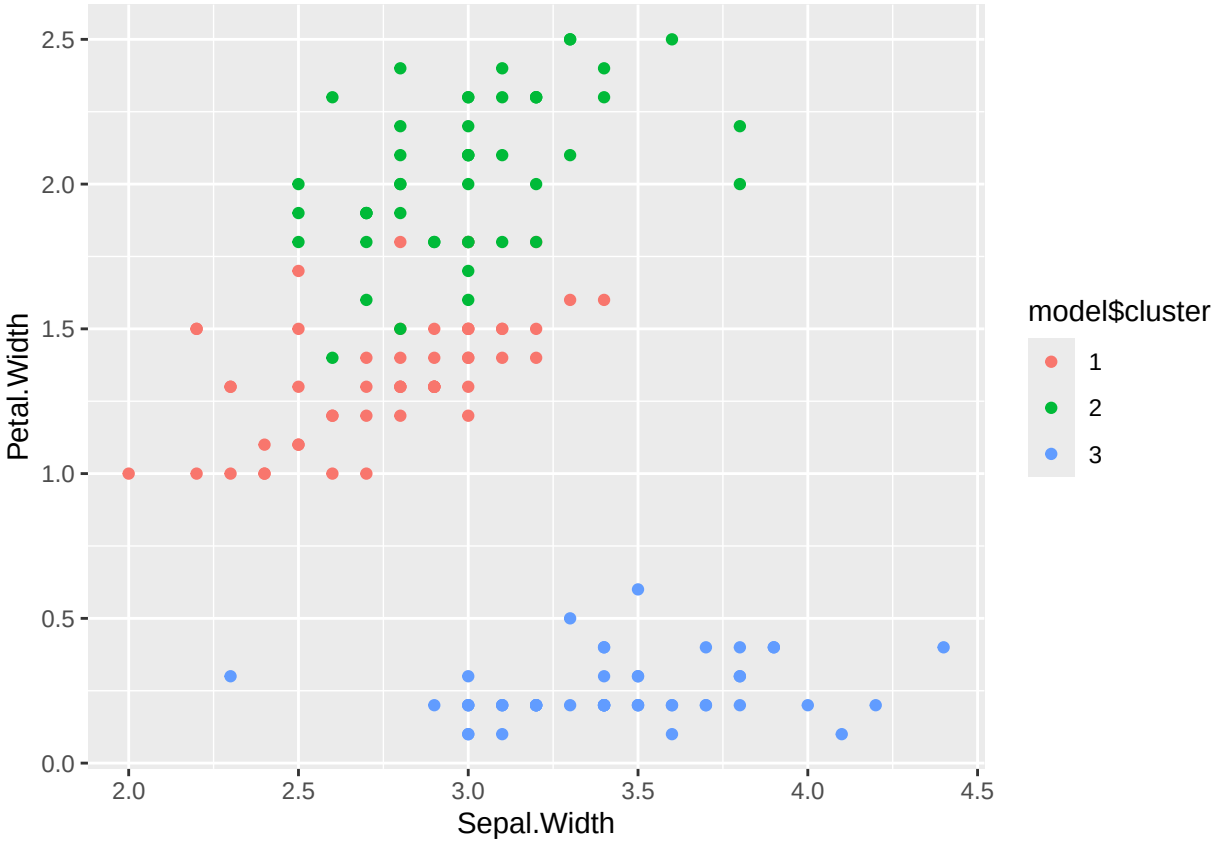
```
model$cluster<-as.factor(model$cluster)
ggplot(data, aes(Petal.Length, Petal.Width, color=model$cluster)) + geom_point()
```



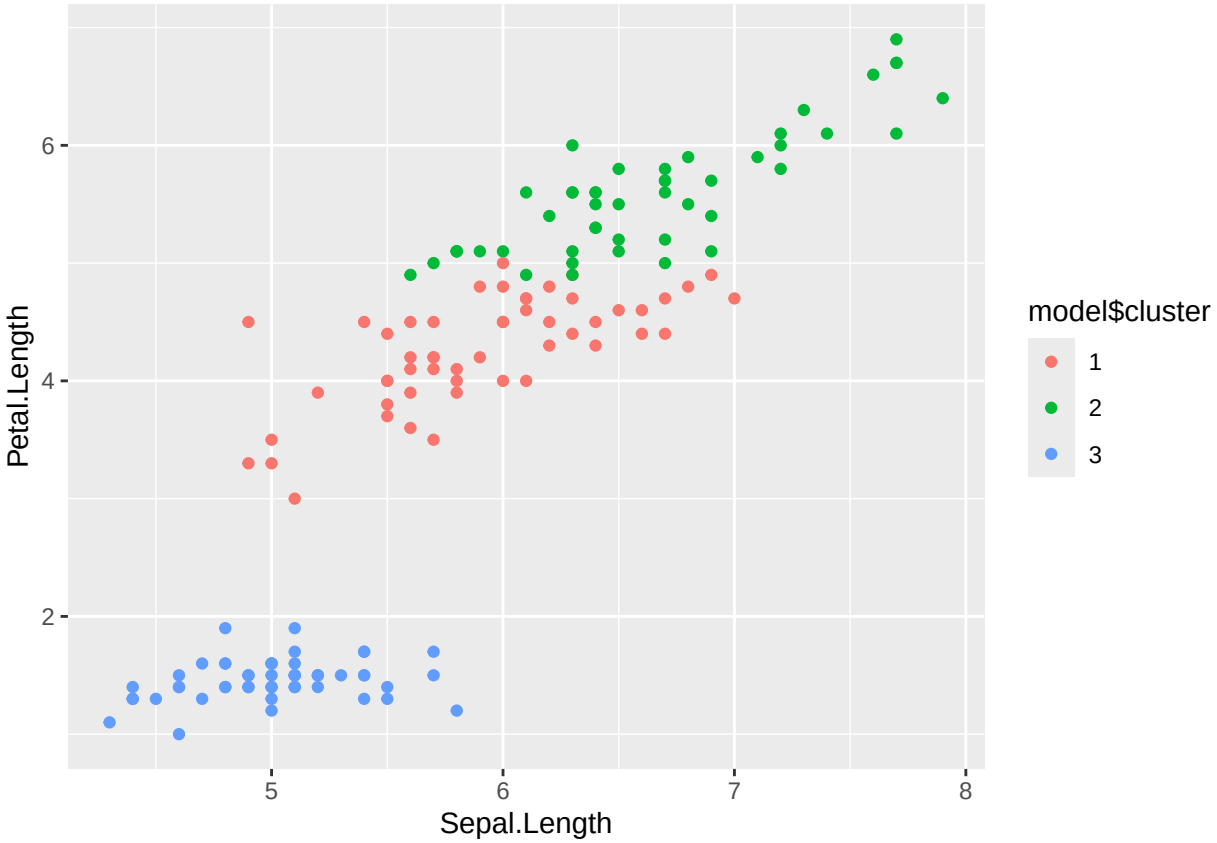

```
ggplot(data, aes(Sepal.Width, Petal.Length, color=model$cluster)) + geom_point()
```



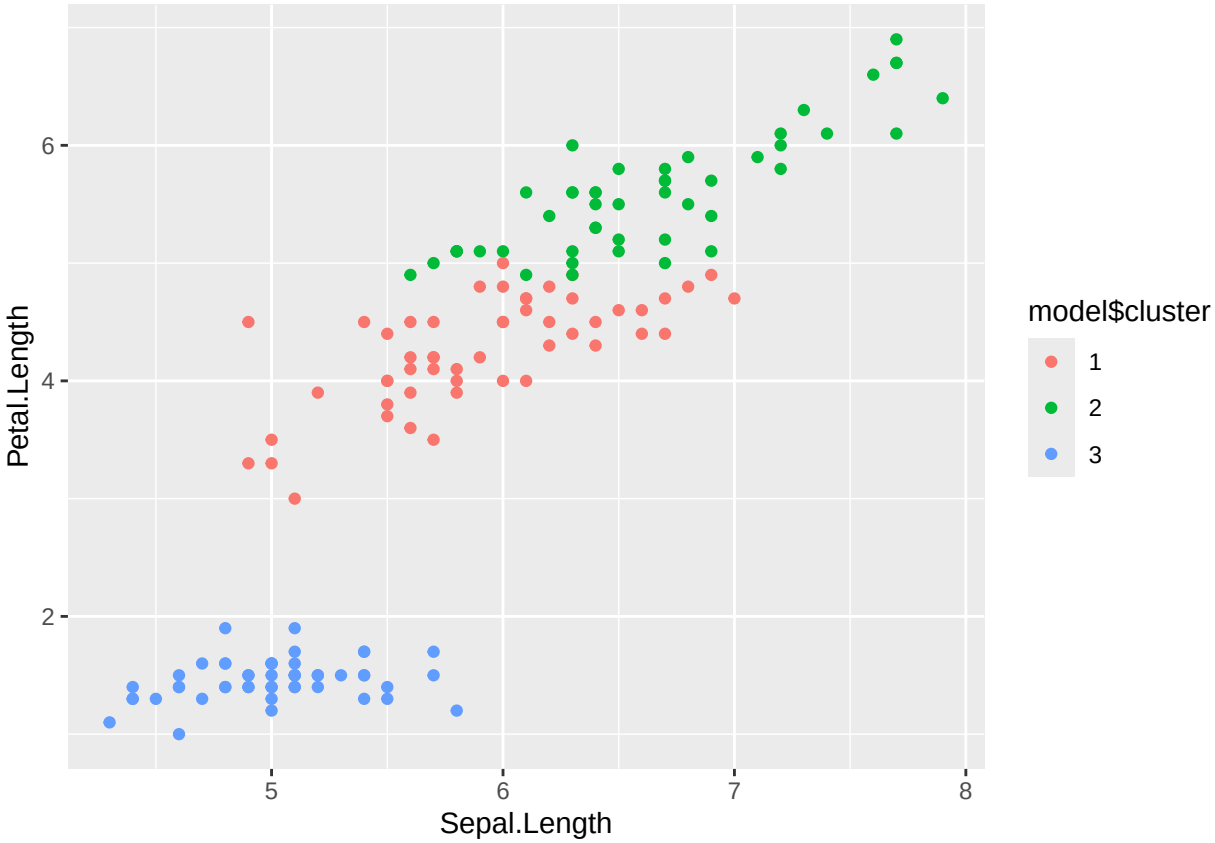

```
ggplot(data, aes(Sepal.Width, Petal.Width, color=model$cluster)) + geom_point()
```



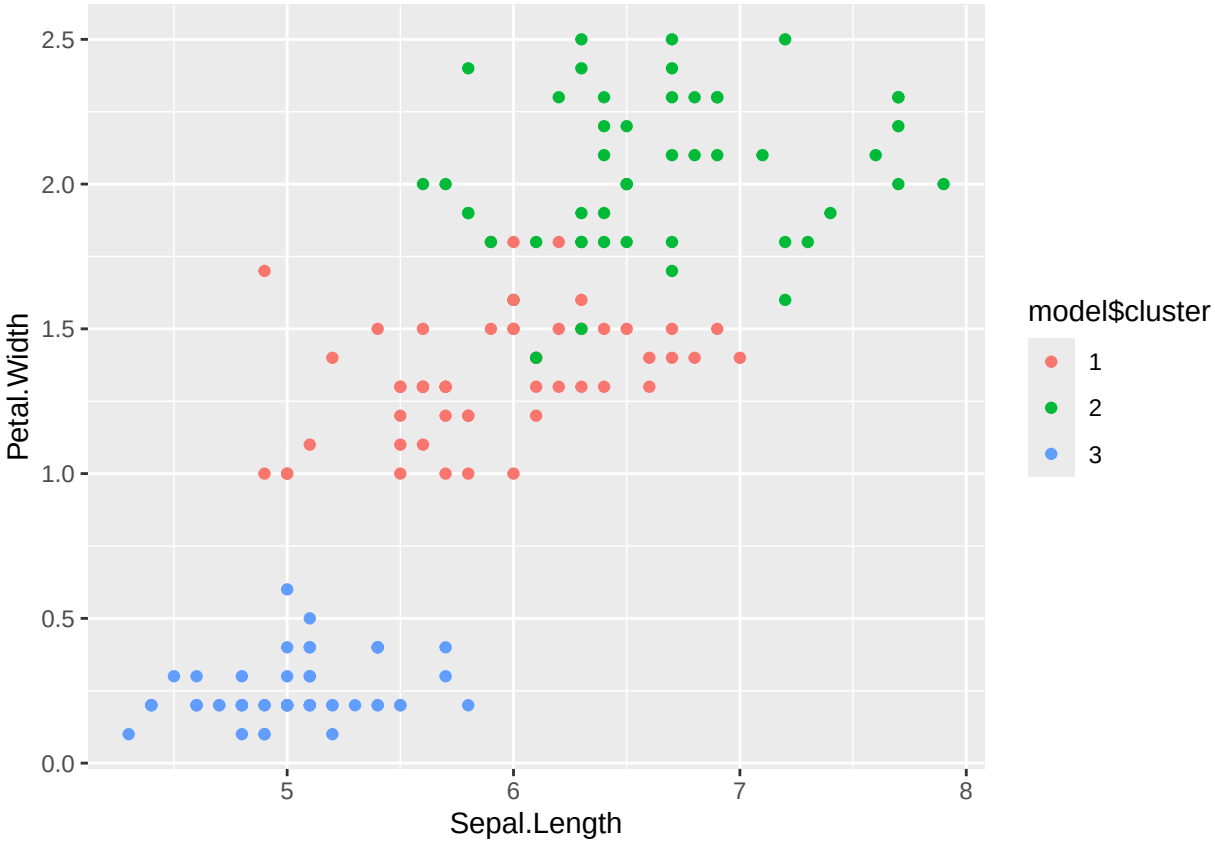

```
ggplot(data, aes(Sepal.Length, Petal.Length, color=model$cluster)) + geom_point()
```




```
ggplot(data, aes(Sepal.Length, Petal.Length, color=model$cluster)) + geom_point()
```




```
ggplot(data, aes(Sepal.Length, Petal.Width, color=model$cluster)) + geom_point()
```


Petal.Width
Sepal.Length
2.5
2.0
1.5
1.0
0.5
0.0
5
6
7
8
model$cluster
1
2
3