Question 4.2

The *iris* data set `iris.txt` contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Iris). *The response values are only given to see how well a specific method performed and should not be used to build the model.*

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.

Answer:

First of all, we need to find the best k value.

Using 3 different ways:

1. Using function: pamk

```
Library()
pamk.best=pamk(data2)
cat('number of clusters:',pamk.best$nc,'\n')
```

the result shows the best k value is 2.

2. Using fuction: NbCluster

```
library(NbClust)
nb_clust <- NbClust(data2, distance='minkowski',min.nc=2,max.nc=10,method = 'kmeans',index='alllong',
alphaBeale=0.1)
```

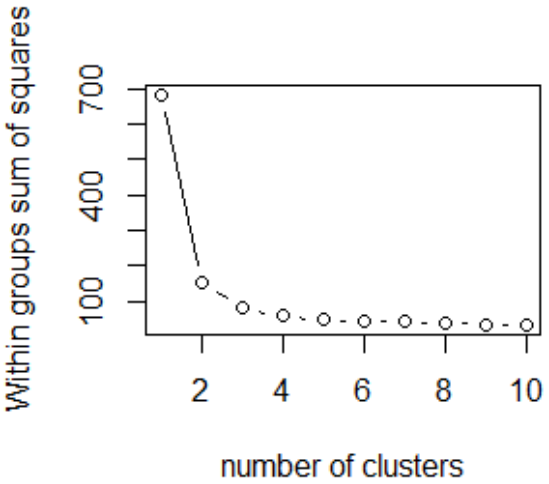
The result shows the best k value is 2.

3. Applying elbow method, calculating sum of squared error.

```
wssplot <- function(data,nc=10,seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data,centers=i)$withinss)
  }
  plot(1:nc,wss,type='b',xlab='number of clusters',
       ylab='Within groups sum of squares' )
}
wssplot(data2)
```

Figure 1 shows the change in sse value when the number of clusters increases.

Figure 1



In this case k=2 or k=3 are both acceptable k values.

Thus we tried both k=2 and k=3 in order to find the best cluster method.

```
model <- kmeans(data2,k_value)
pred <- model$cluster
result_matrix <- matrix(c(data[,5],pred),nrow=150,ncol=2)
```

In order to find out how well our predictor is, we introduce "轮廓系数","簇内平均距离","簇间平均聚类".

```
ave_sil1=km_stats1$avg.silwidth
ave_with1=km_stats1$average.within
ave_bet1=km_stats1$average.between
```

	silwidth	average.within	average.between
k=2	0.6810462	1.2172417	4.022225
k=3	0.5528190	0.9187411	3.386163