1. Use the ISLR libary to get the iris data set. Check the head of the iris Data Frame.

```
install.packages('ISLR')
library(ISLR)
head(str(data.frame(iris)))
```

2. Standardize Data

   Use scale() to standardize the feature columns of the iris dataset. Set this standardized version of the data as a new variable.

```
standardized.feature<-scale(iris[1:4])
```

3. Check that the scaling worked by checking the variance of one of the new columns.

   Output: 1

```
var(standardized.feature[,1])
var(standardized.feature[,2])
```

4. Join the standardized data with the response/target/label column (the column with the species names.

```
final.data<-cbind(standardized.feature,iris[5])
head(final.data)
```

5. Train and Test Splits

   Use the caTools library to split your standardized data into train and test sets. Use a 70/30 split.

```
library(caTools)
set.seed(101)
sample<-sample.split(final.data$Species, SplitRatio = 0.70)
train<-subset(final.data,sample==TRUE)
test<-subset(final.data, sample==FALSE)
```

6. Build a KNN model.

   Call the class library:

   library(class)

```
library(class)
```

7. Use the knn function to predict Species of the test set. Use k=1

```
predicted.species<-knn(train[1:4],test[1:4],train$Species, k=1)
predicted.species
```

8. What was your misclassification rate?

   Output: 0.04444444

   ```
   mean(test$Species!=predicted.species)
   ```

9. Create a plot of the error (misclassification) rate for k values ranging from 1 to 10.

   (Hint): Use the following function for the first part of question:

   ```
   predicted.species <- NULL
   error.rate <- NULL

   for(i in 1:10){
     set.seed(101)
    predicted.species <- knn(train[1:4],test[1:4],train$Species,k=i)
    error.rate[i] <- mean(test$Species != predicted.species)
   }
   ```

   Notice that the error drops to its lowest for k values between 2-6. Then it begins to jump back up again, this is due to how small the data set it. At k=10 you begin to approach setting k=10% of the data, which is quite large.

   ```
   predicted.species<-NULL
   error.rate<-NULL
   for(i in 1:10){
     set.seed(101)
     predicted.species<-knn(train[1:4],test[1:4],train$Species, k=i)
     error.rate[i]<-mean(test$Species!=predicted.species)
   }

   library(ggplot2)
   k.values<-1:10
   error.df<-data.frame(error.rate,k.values)
   pl<-ggplot(error.df,aes(x=k.values,y=error.rate)) + geom_point()
   pl + geom_line(lty="dotted",color='red')
   ```