Chapter 4 Classification

Jishen Yin

2020/5/6

```
knitr::opts_chunk$set(echo = TRUE)
library(ISLR)
library(MASS)
library(class)
library(tidyverse)
library(GGally)
library(gridExtra)
library(grid)
```

Problem 10

This question should be answered using `Weekly` data set, which is part of the `ISLR` package.

```
data(Weekly)
```

(a) Produce some numerical and graphical summaries of the `Weekly` data.

```
summary(Weekly)
```

```
##       Year          Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

(b) Use the full dataset to perform a logistic regression with `Direction` as the response and the five lag variables plus `Volume` as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which one?

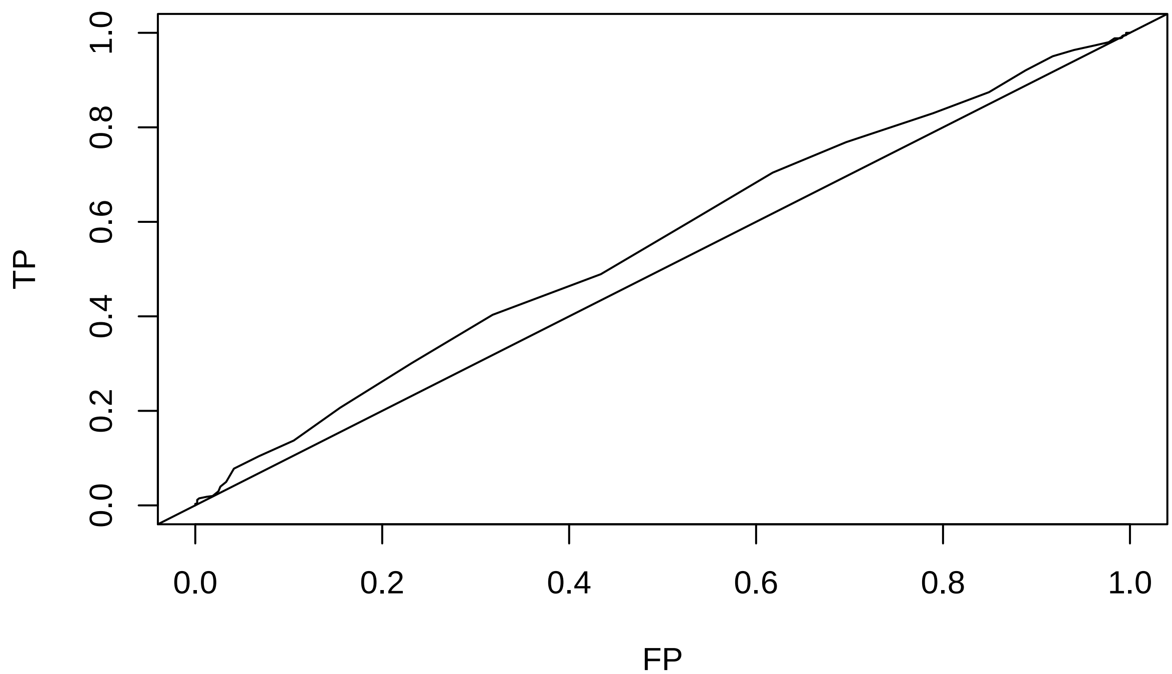
```
lr <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)
summary(lr)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

(c) compute the confusion matrix and overall fraction of correct predictions.

```
pred_prob <- predict(lr, Weekly, type = "response")
thres <- seq(0, 1, 0.01)
TP <- rep(0, 101)
FP <- rep(0, 101)
N <- sum(Weekly$Direction == "Down")
P <- sum(Weekly$Direction == "Up")
for(i in 1:101){
  pred <- pred_prob > thres[i]
  TP[i] <- sum(Weekly$Direction == "Up" & pred)/P
  FP[i] <- sum(Weekly$Direction == "Down" & pred)/N
}
```

```
plot(FP, TP, type = "l", xlim = c(0, 1), ylim = c(0, 1))
abline(a = 0, b = 1)
```




```r
# Choose the best threshold
diff <- TP - FP
thre <- thres[diff == max(diff)]
pred <- ifelse(pred_prob > thre, "Up", "Down")

table(Weekly$Direction, pred)
```

```
##       pred
##        Down  Up
##   Down  185 299
##   Up    179 426
```

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, which `Lag2` as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data.

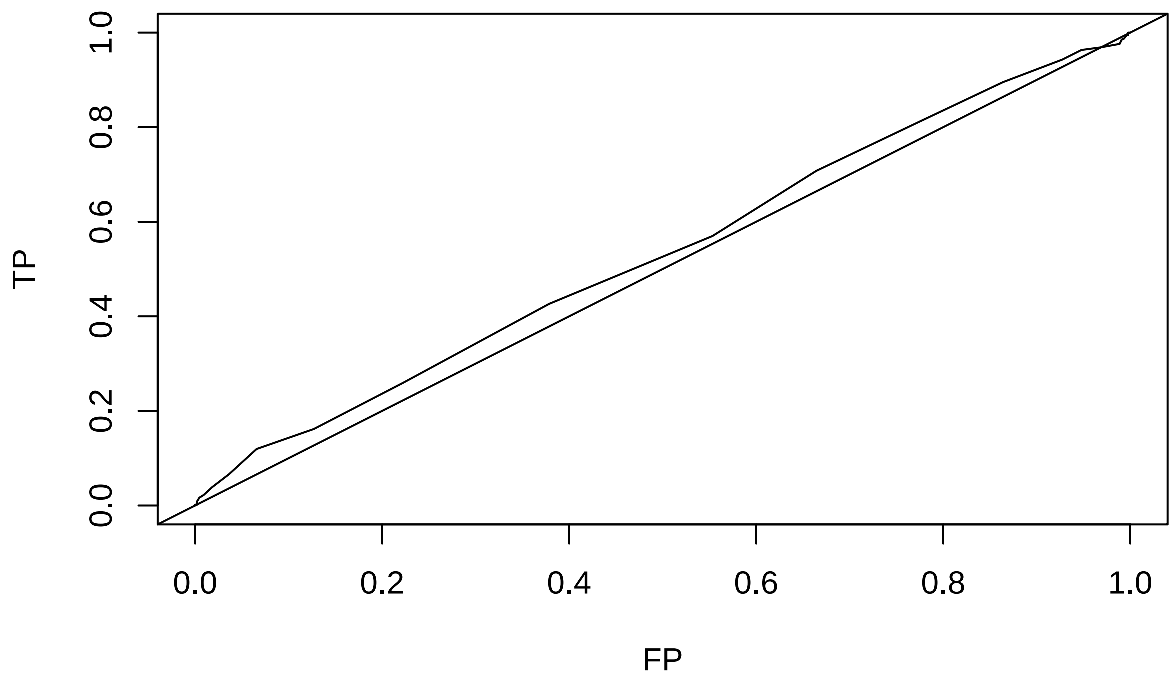
```r
Weekly_train <- Weekly[Weekly$Year <= 2008, ]
Weekly_test <- Weekly[Weekly$Year > 2008, ]

lr_new <- glm(Direction ~ Lag2, data = Weekly_train, family = binomial)
summary(lr_new)
```

```
##
## Call:
```

```
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```r
pred_prob <- predict(lr_new, Weekly_train, type = "response")
thres <- seq(0, 1, 0.01)
TP <- rep(0, 101)
FP <- rep(0, 101)
N <- sum(Weekly_train$Direction == "Down")
P <- sum(Weekly_train$Direction == "Up")
for(i in 1:101){
  pred <- pred_prob > thres[i]
  TP[i] <- sum(Weekly_train$Direction == "Up" & pred)/P
  FP[i] <- sum(Weekly_train$Direction == "Down" & pred)/N
}
```

```r
plot(FP, TP, type = "l", xlim = c(0, 1), ylim = c(0, 1))
abline(a = 0, b = 1)
```




```r
# Choose the best threshold
diff <- TP - FP
thre <- thres[diff == max(diff)]
pred <- ifelse(predict(lr_new, Weekly_test) > thre, "Up", "Down")

table(Weekly_test$Direction, pred)
```

```
##       pred
##        Down Up
##   Down   42  1
##   Up     59  2
```

(e) Repeat (d) using LDA

```r
lda.fit <- lda(Direction ~ Lag2, data = Weekly_train)
lda.pred <- predict(lda.fit, Weekly_test)
lda.class <- lda.pred$class
table(lda.class, Weekly_test$Direction)
```

```
##
## lda.class Down Up
##      Down    9  5
##      Up     34 56
```

(f) Repeat (d) using QDA

```
qda.fit <- qda(Direction ~ Lag2, data = Weekly_train)
qda.pred <- predict(qda.fit, Weekly_test)
qda.class <- qda.pred$class
table(qda.class, Weekly_test$Direction)
```

```
##
## qda.class Down Up
##      Down    0  0
##      Up     43 61
```

(g) Repeat (d) with KNN

```
knn.pred <- knn(data.frame(Lag2 = Weekly_train$Lag2),
                data.frame(Lag2 = Weekly_test$Lag2), Weekly_train$Direction, k = 4)
table(knn.pred, Weekly_test$Direction)
```

```
##
## knn.pred Down Up
##     Down   19 18
##     Up     24 43
```

Problem 11

In this peoblem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set.

(a) Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value above its median, and a 0 if `mpg` contains a value below its median.

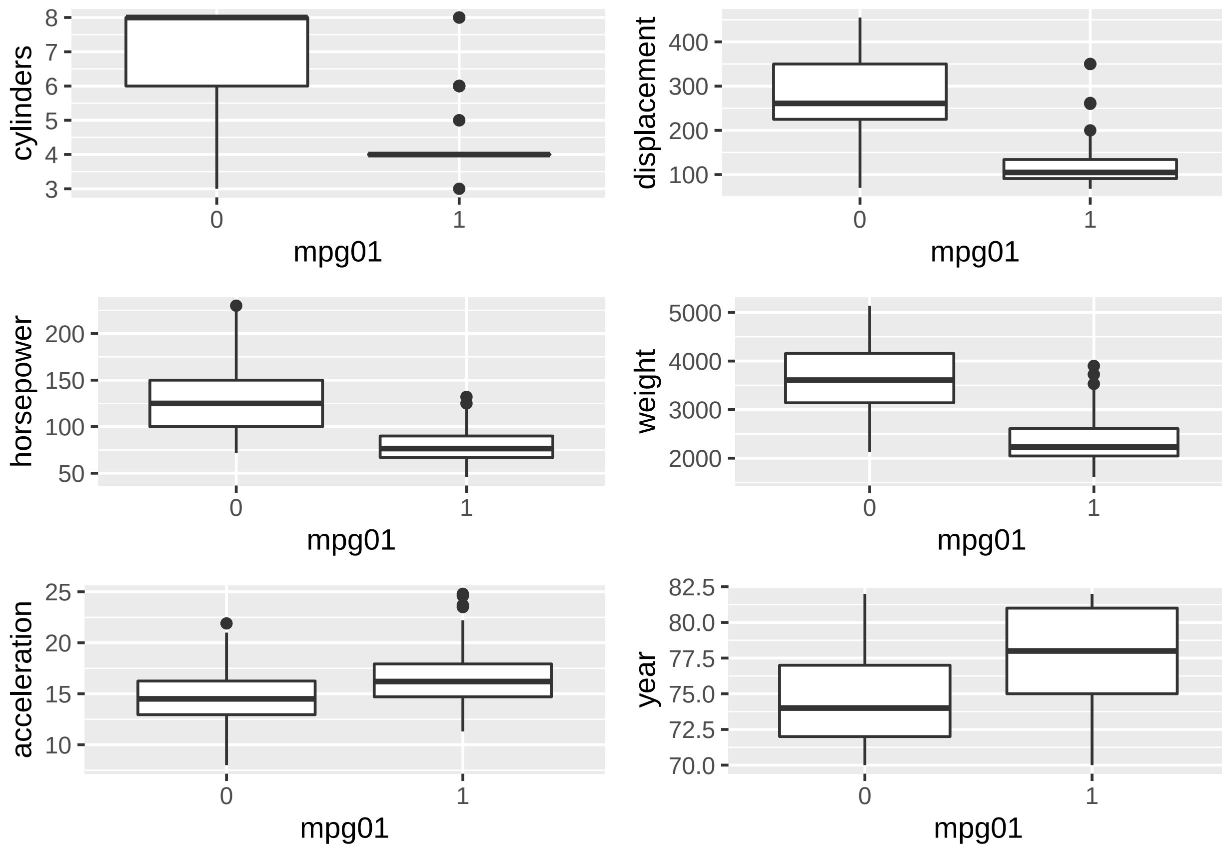
```
data(Auto)
med <- median(Auto$mpg)
Auto <- Auto %>%
  mutate(mpg01 = ifelse(mpg > med, 1, 0)) %>%
  select(-mpg, -name)
Auto$mpg01 = as.factor(Auto$mpg01)
Auto$origin = as.factor(Auto$origin)
```

(b) Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features. Which of the other features seem most likely to be useful in predicting `mpg01`?

```
table(Auto$origin, Auto$mpg01)
```

```
##
##       0   1
##   1 173  72
##   2  14  54
##   3   9  70
```

```r
g1 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = cylinders, group = mpg01))
g2 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = displacement, group = mpg01))
g3 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = horsepower, group = mpg01))
g4 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = weight, group = mpg01))
g5 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = acceleration, group = mpg01))
g6 <- ggplot(data = Auto) +
  geom_boxplot(aes(x = mpg01, y = year, group = mpg01))
grid.arrange(g1, g2, g3, g4, g5, g6, nrow = 3)
```



(c) Split the data into a training set and a test set

```r
Auto_train <- Auto[1:300, ]
Auto_test <- Auto[301:392, ]
```

(d) Perform LDA on the training data in order to predict `mpg01` using the variables that seemed most assciated with `mpg01`.

```r
lda.fit <- lda(mpg01 ~ .-year, data = Auto_train)
lda.pred <- predict(lda.fit, Auto_test)
lda.class <- lda.pred$class
mean(lda.class == Auto_test$mpg01)
```

```
## [1] 0.8695652
```

(e) Perform QDA on the training data in order to predict `mpg01` using the variables that seemed most assciated with `mpg01`.

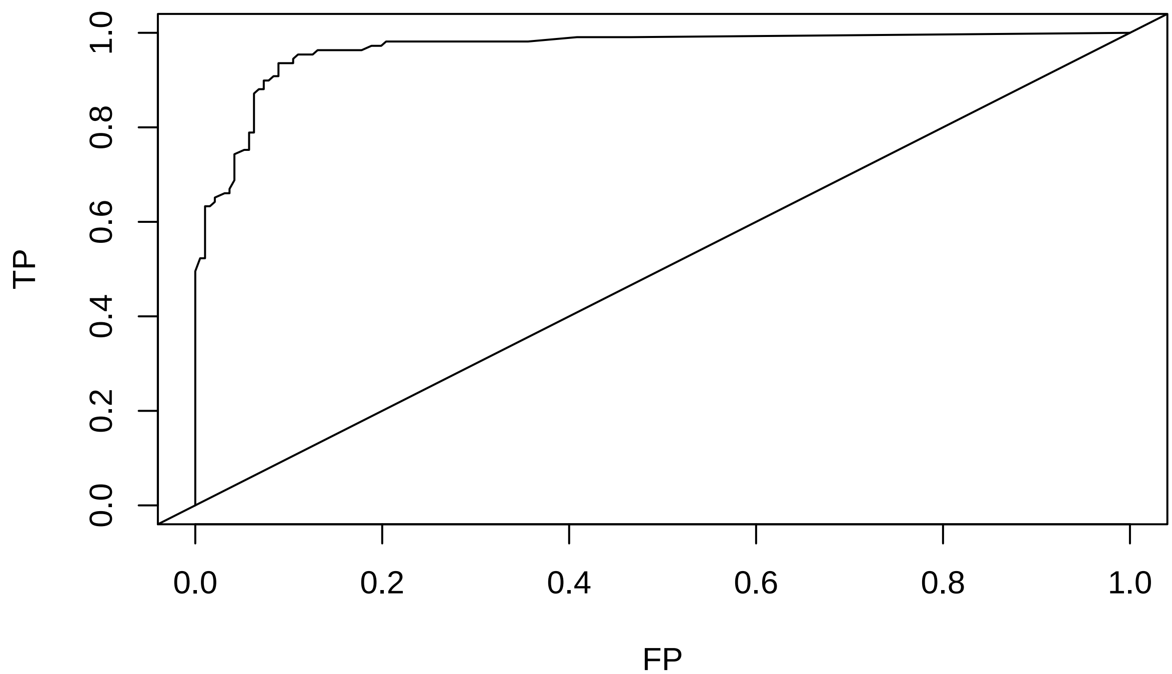
```r
qda.fit <- qda(mpg01 ~ .-year, data = Auto_train)
qda.pred <- predict(qda.fit, Auto_test)
qda.class <- qda.pred$class
mean(qda.class == Auto_test$mpg01)
```

```
## [1] 0.8804348
```

(e) Perform Logistic Regression on the training data in order to predict `mpg01` using the variables that seemed most assciated with `mpg01`.

```r
lr.fit <- glm(mpg01 ~ .-year, data = Auto_train, family = binomial)

pred_prob <- predict(lr.fit, Auto_train, type = "response")
thres <- seq(0, 1, 0.01)
TP <- rep(0, 101)
FP <- rep(0, 101)
N <- sum(Auto_train$mpg01 == 0)
P <- sum(Auto_train$mpg01 == 1)
for(i in 1:101){
  pred <- pred_prob > thres[i]
  TP[i] <- sum(Auto_train$mpg01 == 1 & pred)/P
  FP[i] <- sum(Auto_train$mpg01 == 0 & pred)/N
}
```

```r
plot(FP, TP, type = "l", xlim = c(0, 1), ylim = c(0, 1))
abline(a = 0, b = 1)
```



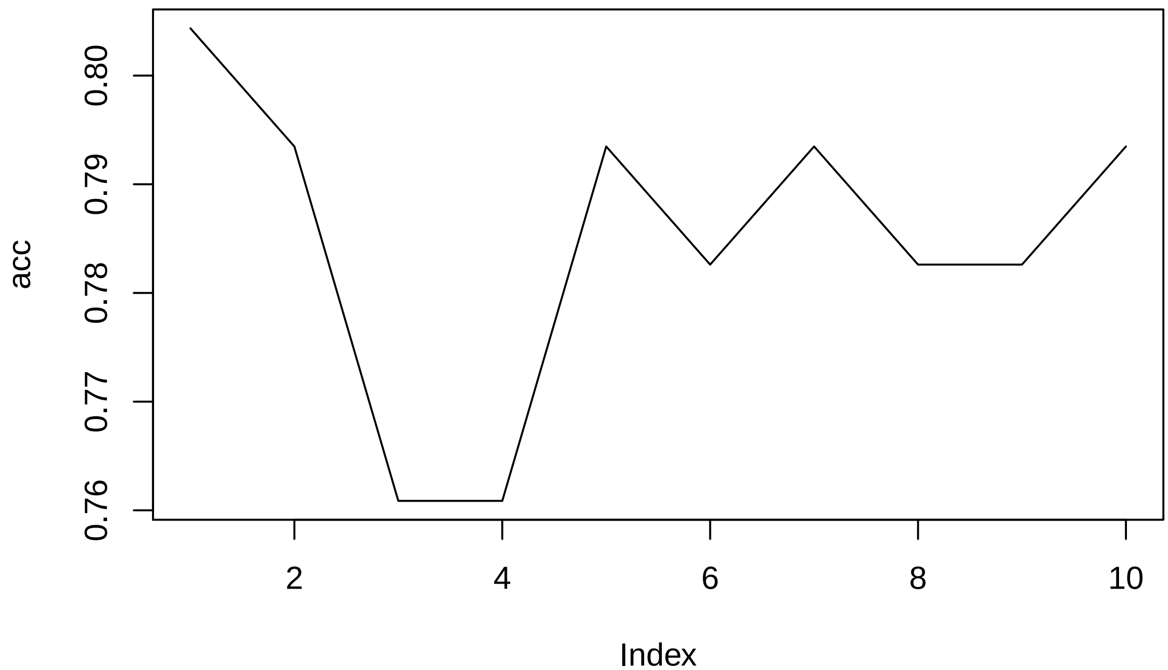

```r
# Choose the best threshold
diff <- TP - FP
thre <- thres[diff == max(diff)][1]
pred <- ifelse(predict(lr.fit, Auto_test) > thre, 1, 0)

mean(Auto_test$mpg01 == pred)
```

```
## [1] 0.7065217
```

(g) perform KNN on the training data, with several values of K.

```r
X_train <- Auto_train[c(1:5, 7)]
X_test <- Auto_test[c(1:5, 7)]
y_train <- Auto_train$mpg01
y_test <- Auto_test$mpg01

acc <- sapply(1:10, function(x){
  knn.pred <- knn(X_train, X_test, y_train, k = x)
  return(mean(knn.pred == y_test))
})
```

```r
plot(acc, type = "l")
```


acc
0.76
0.77
0.78
0.79
0.80
2
4
6
8
10
Index