MA679 Classification homework

Jiachen Feng

2021/2/6

4.6

(a)

```r
p <- plogis(-6+0.05*40+3.5)
p
```

```
## [1] 0.3775407
```

$$P(Y = 1) = logit^{-1}(-6 + 0.05 * 40 + 3.5) = 37.75\%$$

(b)

$$0.5 = logit^{-1}(-6 + 0.05 * t + 3.5)$$
$$t = 50h$$

4.8

Under this circumstance, QDA performs best. The test error rate using logistic regression is higher than KNN-1 test, which means the responses from the logistic function using quadratic variable as predictors. Consequently, there is a quadratic decision boundary. Therefore, QDA performs best.

4.9

(a)

$$\frac{P(default)}{1 - P(default)} = 0.37$$
$$P(default) = 0.27$$

(b)

$$odds = \frac{0.16}{1 - 0.16} = 0.19$$

4.10

(a)

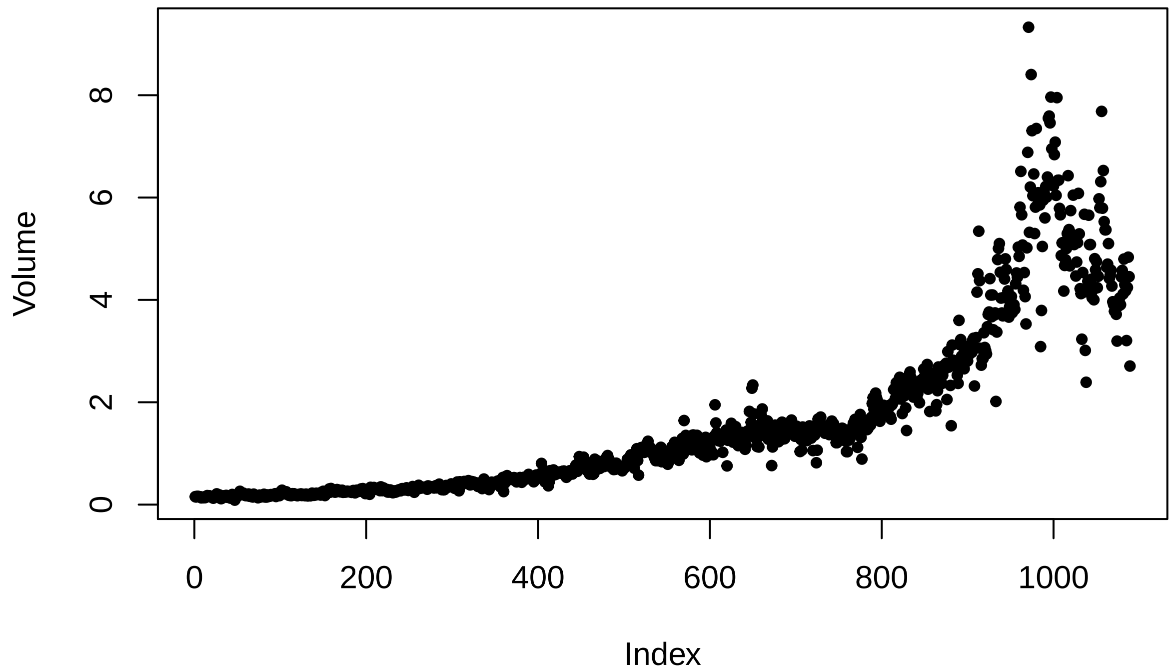
```r
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

```r
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
##
##
##
##
```

```r
attach(Weekly)
plot(Volume,pch=20)
```



The max and min values of the **lag** variables are the same.

(b)

```r
fit_1 <- glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Weekly,family = binomial)
summary(fit_1)
```

```
##
## 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
```

Lag2 appears to be a significant predictor.

(c)

```r
pred <- predict(fit_1, type = "response")
contrasts(Direction)
```

```
##      Up
## Down  0
## Up    1
```

```r
pred1 <- rep("Down", length(pred))
pred1[pred > 0.5] <- "Up"

# Confusion Matrix
table(pred1, Direction)
```

```
##       Direction
## pred1  Down  Up
##   Down   54  48
##   Up    430 557
```

```r
# Compute overall fraction of correct predictions
mean(pred1==Direction)
```

```
## [1] 0.5610652
```

Training error rate is high, and this method needs to be improved.

(d)

```r
train <- filter(Weekly,Year<=2008)
heldout <- filter(Weekly,Year>2008)
fit_2 <- glm(Direction~Lag2,data = train,family = binomial)
pred <- predict(fit_2,heldout,type="response")

pred2 <- rep("Down", length(pred))
pred2[pred > 0.5] <- "Up"

# Confusion Matrix
table(pred2, heldout$Direction)
```

```
##
## pred2  Down Up
##   Down    9  5
##   Up     34 56
```

```r
# Compute overall fraction of correct predictions
mean(pred2==heldout$Direction)
```

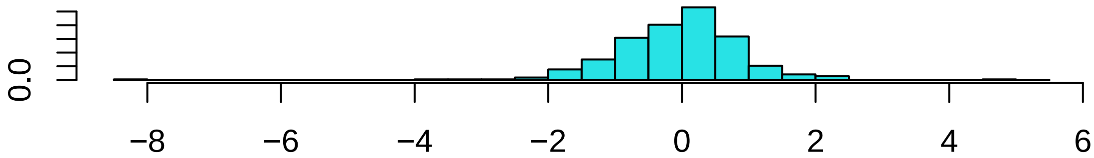
```
## [1] 0.625
```

(e)

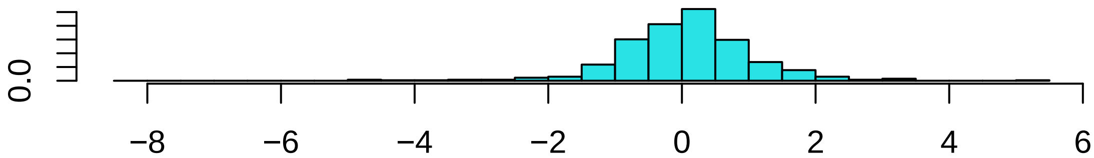
```r
fit_3 <- lda(Direction~Lag2,data = train)
fit_3
```

```
## Call:
## lda(Direction ~ Lag2, data = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```r
plot(fit_3)
```



group Down



group Up

```r
pred <- predict(fit_3,heldout,type="response")
names(pred)
```

```
## [1] "class"     "posterior" "x"
```

```r
# Confusion Matrix
table(pred$class, heldout$Direction)
```

```
##
##        Down Up
##   Down    9  5
##   Up     34 56
```

```r
# Compute overall fraction of correct predictions
mean(pred$class==heldout$Direction)
```

```
## [1] 0.625
```

(f)

```r
fit_4 <- qda(Direction~Lag2,data = train)
fit_4
```

```
## Call:
## qda(Direction ~ Lag2, data = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
```

```
##
## Group means:
##              Lag2
## Down -0.03568254
## Up    0.26036581
```

```r
pred <- predict(fit_4,heldout,type="response")
names(pred)
```

```
## [1] "class"     "posterior"
```

```r
# Confusion Matrix
table(pred$class, heldout$Direction)
```

```
##
##         Down Up
##   Down     0  0
##   Up      43 61
```

```r
# Compute overall fraction of correct predictions
mean(pred$class==heldout$Direction)
```

```
## [1] 0.5865385
```

(g)

```r
train <- filter(Weekly,Year<=2008)
train <- train[,3]
heldout <- filter(Weekly,Year>2008)
heldout <- heldout[,3]

direction <- filter(Weekly,Year<=2008)$Direction
train <- as.matrix(na.omit(train))
test <- as.matrix(na.omit(heldout))

set.seed(1)
predknn <- knn(train,test,direction,k=1)
table(predknn, filter(Weekly,Year>2008)$Direction)
```

```
##
## predknn Down Up
##    Down   21 30
##    Up     22 31
```

```r
mean(predknn==filter(Weekly,Year>2008)$Direction)
```

```
## [1] 0.5
```

(h) LDA and logistic regression provide the best results.

(i)

```r
train <- filter(Weekly,Year<=2008)
heldout <- filter(Weekly,Year>2008)

#Logistic regression
fit_5 <- glm(Direction~Lag2^2,data = train,family = binomial)
pred <- predict(fit_5,heldout,type="response")

pred3 <- rep("Down", length(pred))
```

```r
pred3[pred > 0.5] <- "Up"

## Confusion Matrix
table(pred3, heldout$Direction)
```

```
##
## pred3  Down Up
##   Down    9  5
##   Up     34 56
```

```r
## Compute overall fraction of correct predictions
mean(pred3==heldout$Direction)
```

```
## [1] 0.625
```

```r
#LDA
fit_6 <- lda(Direction~Lag2^2,data = train)

pred <- predict(fit_6,heldout,type="response")

## Confusion Matrix
table(pred$class, heldout$Direction)
```

```
##
##        Down Up
##   Down    9  5
##   Up     34 56
```

```r
## Compute overall fraction of correct predictions
mean(pred$class==heldout$Direction)
```

```
## [1] 0.625
```

```r
#QDA
fit_7 <- qda(Direction~Lag2^2,data = train)

pred <- predict(fit_7,heldout,type="response")

## Confusion Matrix
table(pred$class, heldout$Direction)
```

```
##
##        Down Up
##   Down    0  0
##   Up     43 61
```

```r
## Compute overall fraction of correct predictions
mean(pred$class==heldout$Direction)
```

```
## [1] 0.5865385
```

```r
#KNN-3
train <- train[,3]^2
heldout <- heldout[,3]^2

direction <- filter(Weekly,Year<=2008)$Direction
train <- as.matrix(na.omit(train))
test <- as.matrix(na.omit(heldout))
```

```r
set.seed(1)
predknn <- knn(train,test,direction,k=3)
table(predknn, filter(Weekly,Year>2008)$Direction)
```

```
##
## predknn Down Up
##     Down   22 28
##     Up     21 33
```

```r
mean(predknn==filter(Weekly,Year>2008)$Direction)
```

```
## [1] 0.5288462
```

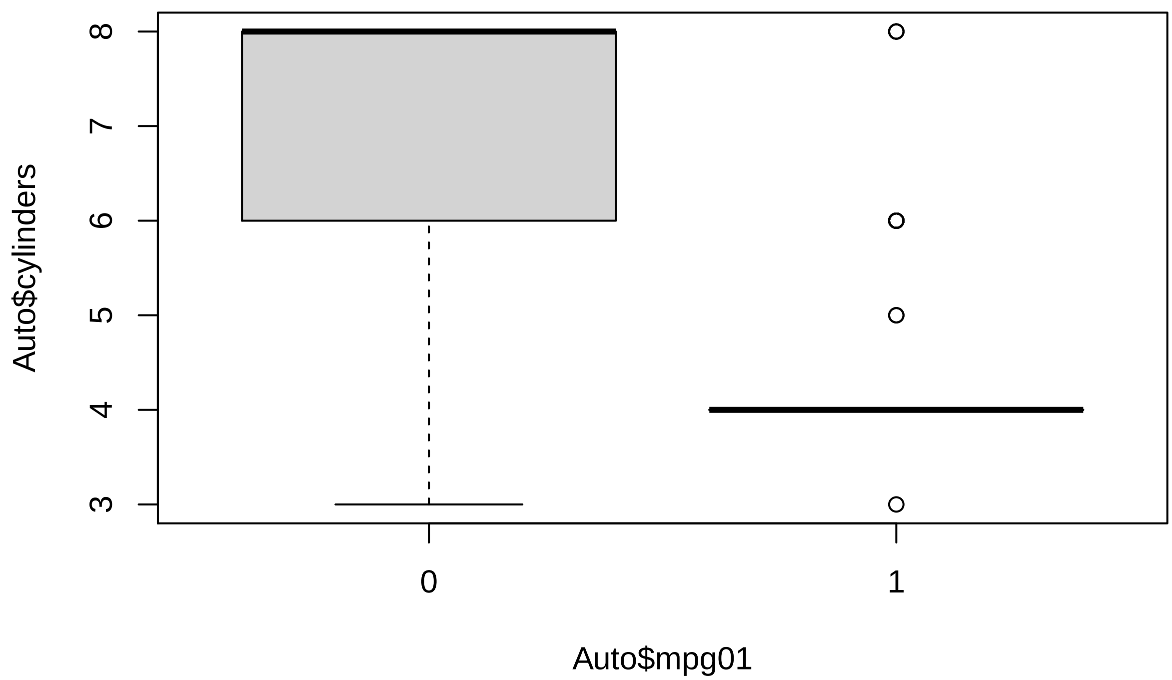
4.11

(a)

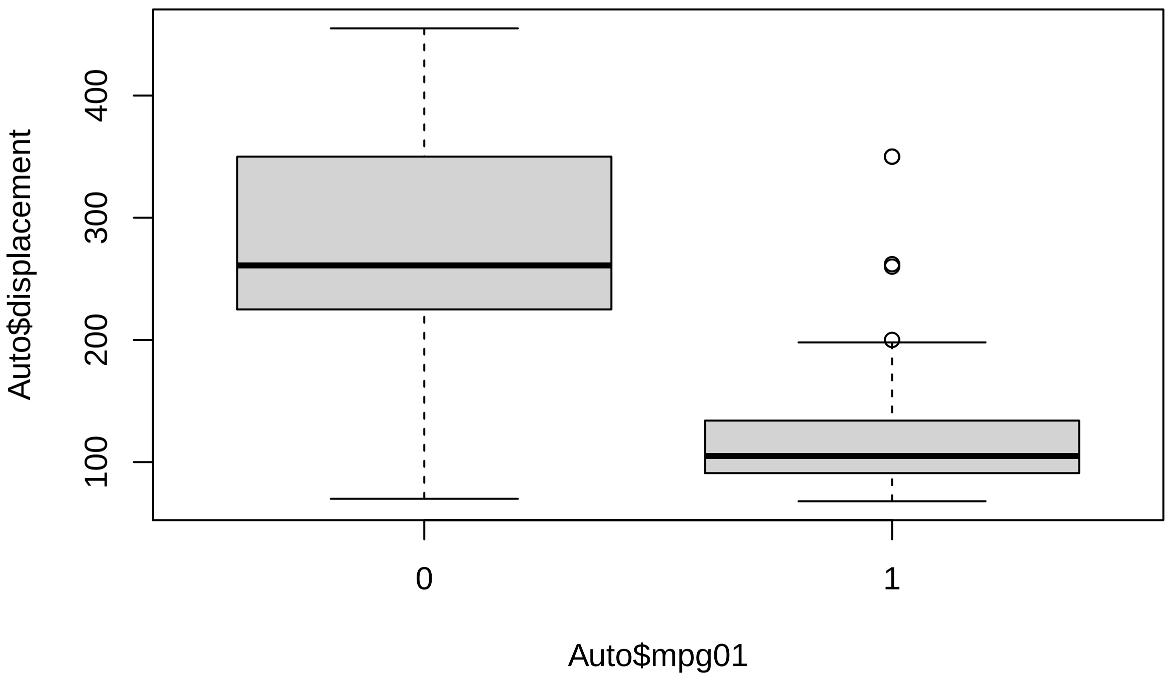
```r
Auto <- data.frame(Auto)
Auto$mpg01[Auto$mpg>median(Auto$mpg)] <- 1
Auto$mpg01[Auto$mpg<median(Auto$mpg)] <- 0
```

(b)

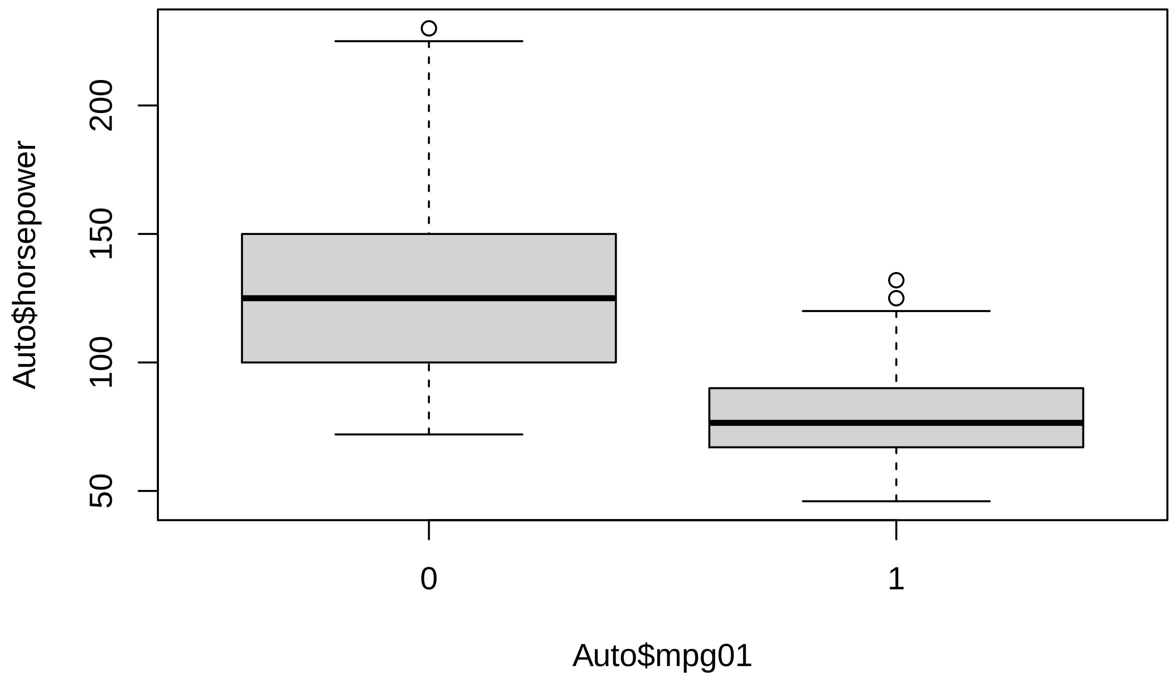
```r
boxplot(Auto$cylinders~Auto$mpg01)
```



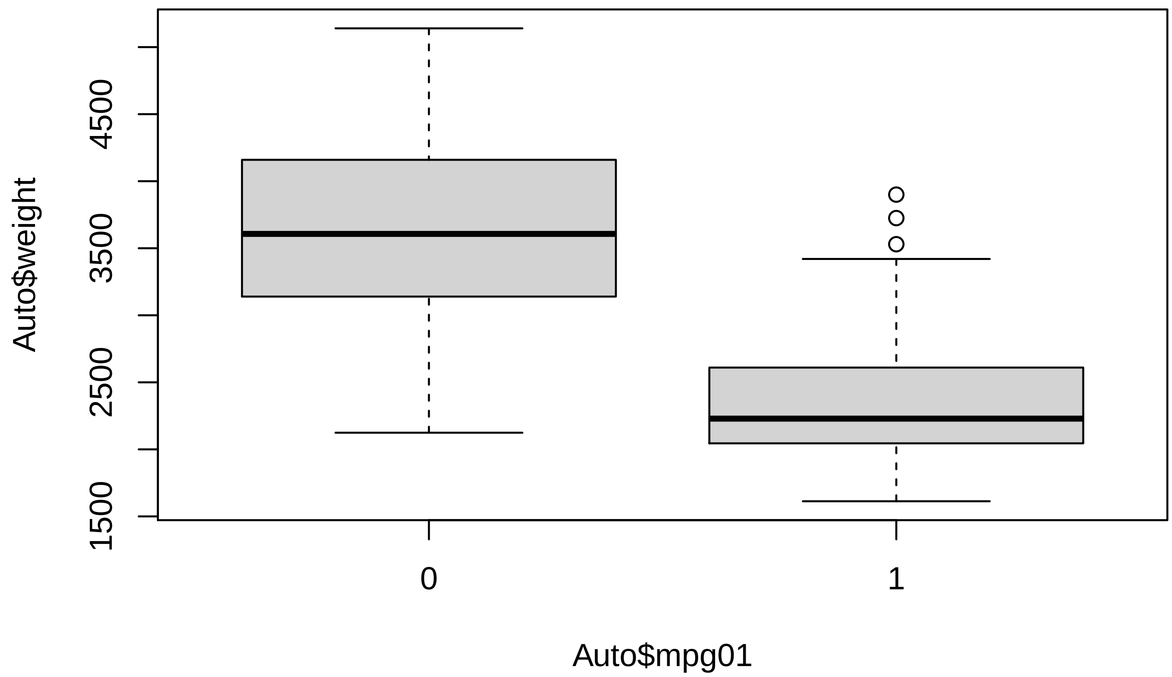

```r
boxplot(Auto$displacement~Auto$mpg01)
```



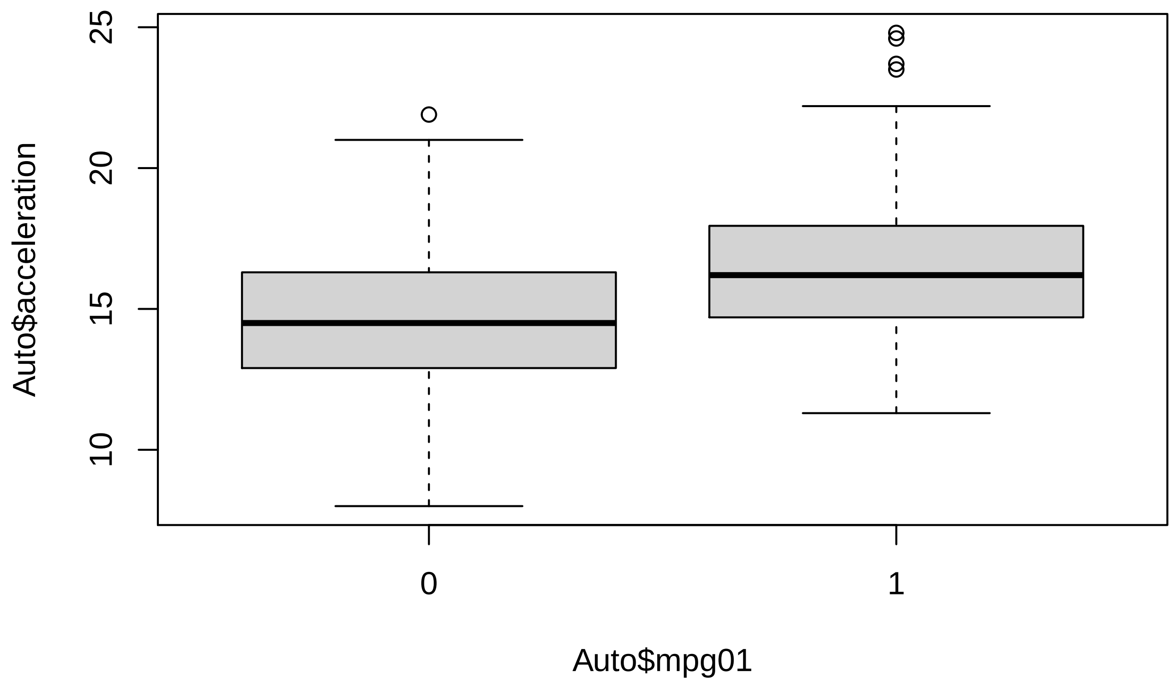

```
boxplot(Auto$horsepower~Auto$mpg01)
```



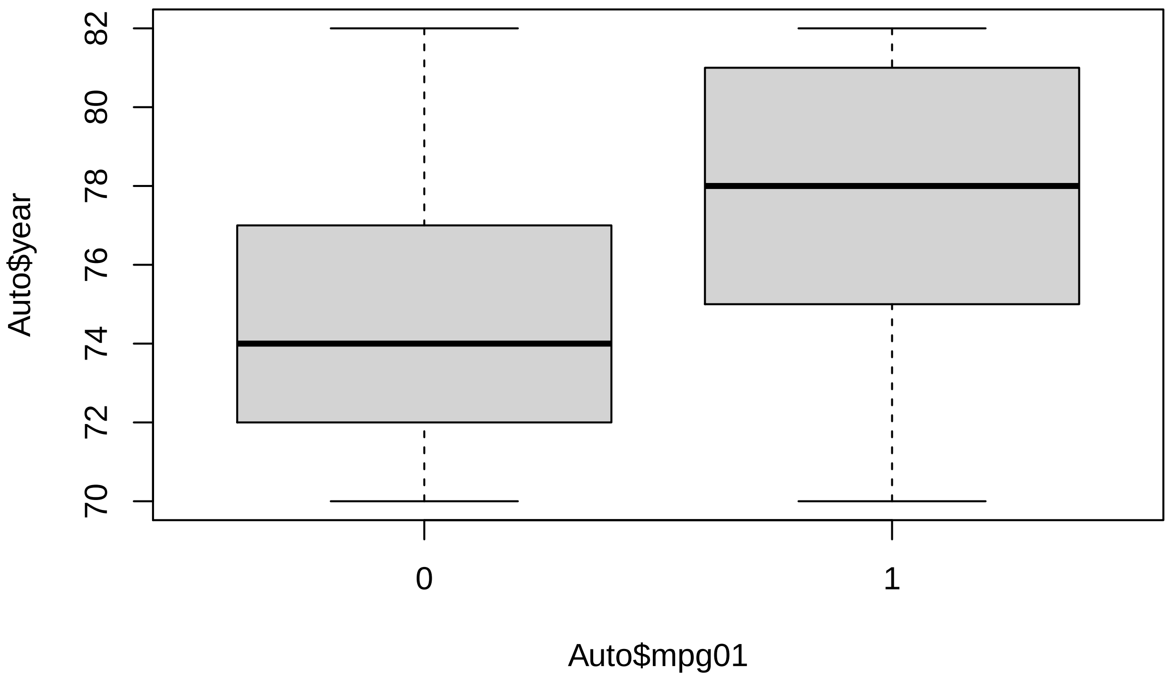

```
boxplot(Auto$weight~Auto$mpg01)
```



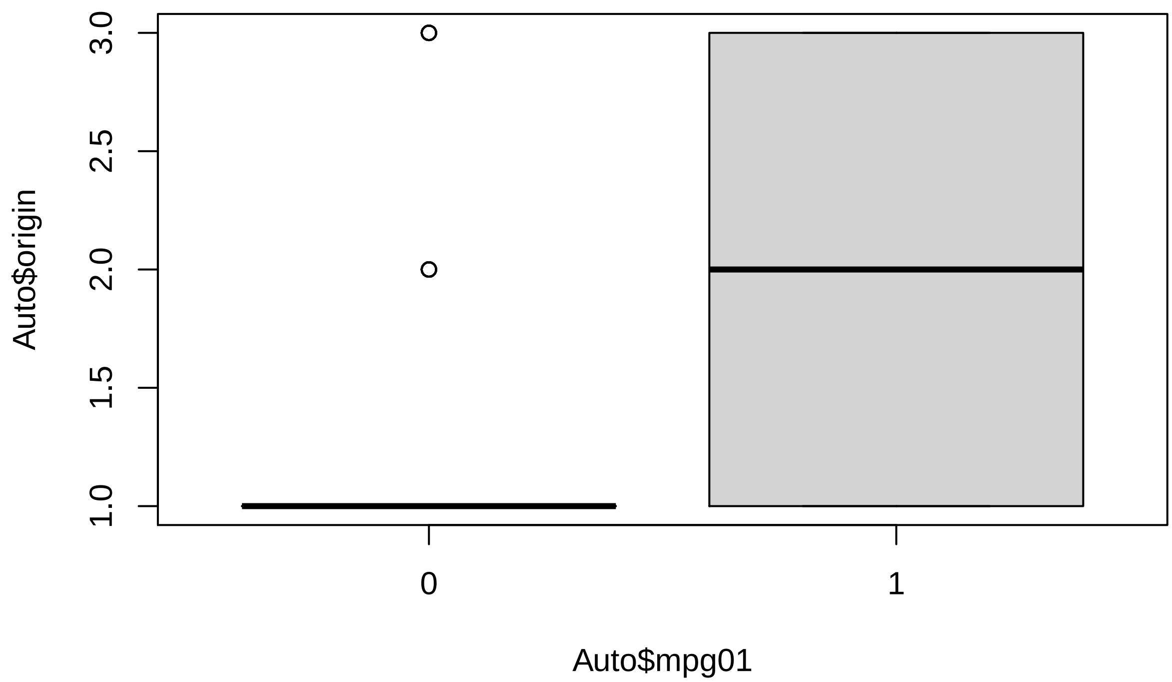

```r
boxplot(Auto$acceleration~Auto$mpg01)
```




```
boxplot(Auto$year~Auto$mpg01)
```




```
boxplot(Auto$origin~Auto$mpg01)
```



Acceleration seems to be an important feature.

(c)

```
train <- filter(Auto,year<=78)
test <- filter(Auto,year<78)
```

(d)

```
fit_8 <- lda(mpg01~acceleration,data = train)

pred <- predict(fit_8,test,type="response")

## Confusion Matrix
table(pred$class, test$mpg01)
```

```
##
##       0   1
##   0 133  60
##   1  24  25
```

```
## Compute overall fraction of correct predictions
mean(pred$class==test$mpg01)
```

```
## [1] 0.6528926
```

(e)

```
fit_9 <- qda(mpg01~acceleration,data = train)
```

```r
pred <- predict(fit_9,test,type="response")

## Confusion Matrix
table(pred$class, test$mpg01)
```

```
##
##       0   1
##   0 129  51
##   1  28  34
```

```r
## Compute overall fraction of correct predictions
mean(pred$class==test$mpg01)
```

```
## [1] 0.6735537
```

(f)

```r
fit_10 <- glm(mpg01~acceleration,data = train,family = binomial)

pred <- predict(fit_10,test,type="response")
pred1 <- rep("0", length(pred))
pred1[pred > 0.5] <- "1"

## Confusion Matrix
table(pred1, test$mpg01)
```

```
##
## pred1   0   1
##     0 133  60
##     1  24  25
```

```r
## Compute overall fraction of correct predictions
mean(pred1==test$mpg01)
```

```
## [1] 0.6528926
```

(g)

```r
train <- filter(Auto,year<=78)
test <- filter(Auto,year<78)
train <- train[,6]
test <- test[,6]

mpg <- filter(Auto,year<=78)$mpg01
train <- as.matrix(na.omit(train))
test <- as.matrix(na.omit(test))

set.seed(1)
#k=1
predknn <- knn(train,test,mpg,k=1)
## mean(predknn==filter(Auto,year>78)$mpg01)

#k=5
predknn <- knn(train,test,mpg,k=5)
## mean(predknn==filter(Auto,year>78)$mpg01)

#k=15
```

```r
predknn <- knn(train,test,mpg,k=15)
## mean(predknn==filter(Auto,year>78)$mpg01)

#k=50
predknn <- knn(train,test,mpg,k=50)
## mean(predknn==filter(Auto,year>78)$mpg01)
```

When K=50, KNN performs the best.

4.12

(a)

```r
Power <- function(){print(2^3)}
Power()
```

```
## [1] 8
```

(b)

```r
Power2 <- function(x,a){print(x^a)}
Power2(3,8)
```

```
## [1] 6561
```

(c)

```r
Power2(10,3)
```

```
## [1] 1000
```

```r
Power2(8,17)
```

```
## [1] 2.2518e+15
```

```r
Power2(131,3)
```

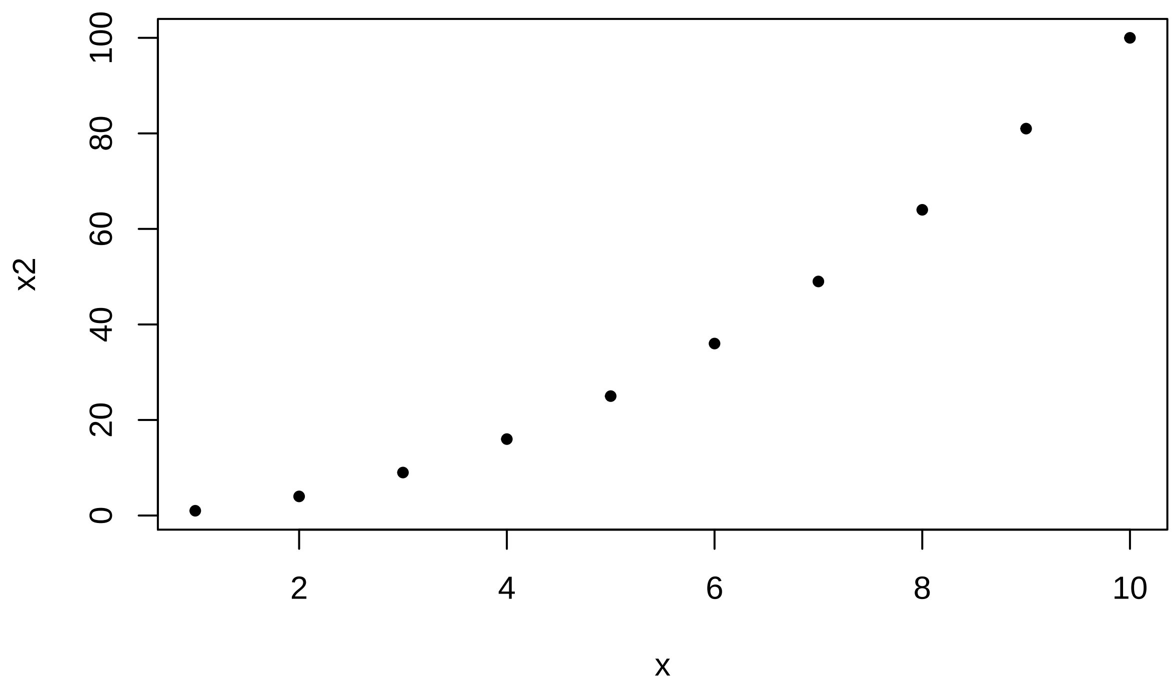
```
## [1] 2248091
```

(d)

```r
Power3 <- function(x,a){return(x^a)}
```

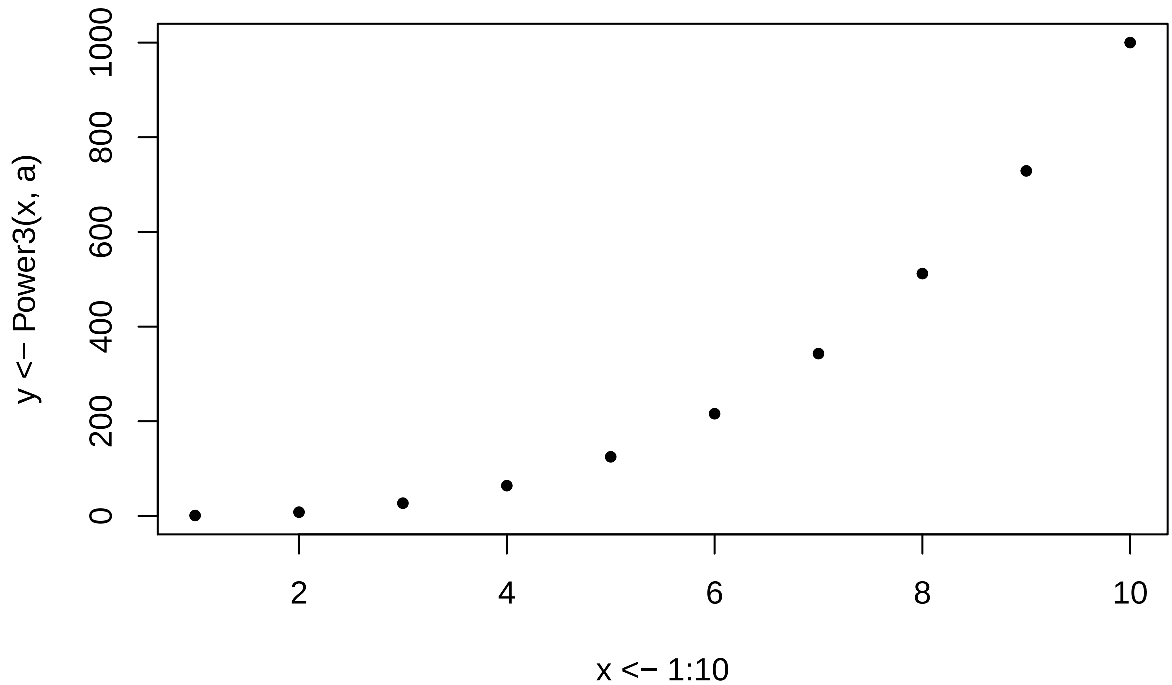
(e)

```r
plot(x <- 1:10, y <- Power3(x,2), xlab="x", ylab="x2",pch=20)
```



(f)

```
PlotPower <- function(x,a){
  plot(x <- 1:10, y <- Power3(x,a),pch=20)
}
PlotPower(1:10,3)
```



4.13

```r
Boston <- data.frame(Boston)
Boston$crime[Boston$crim>median(Boston$crim)] <- 1
Boston$crime[Boston$crim<median(Boston$crim)] <- 0

#LDA
fit_11 <- lda(crime~zn+indus+nox+rm,data = Boston)

pred <- predict(fit_11,Boston,type="response")

## Confusion Matrix
table(pred$class, Boston$crime)
```

```
##
##         0   1
##    0 226  58
##    1  27 195
```

```r
## Compute overall fraction of correct predictions
mean(pred$class==Boston$crime)
```

```
## [1] 0.8320158
```

```r
#logistic Regression
fit_12 <- glm(crime~zn+indus+nox+rm,data = Boston,family = binomial)
```

```r
pred <- predict(fit_12,Boston,type="response")
pred1 <- rep("0", length(pred))
pred1[pred > 0.5] <- "1"

## Confusion Matrix
table(pred1, Boston$crime)
```

```
##
## pred1   0   1
##     0 215  37
##     1  38 216
```

```r
## Compute overall fraction of correct predictions
mean(pred1==Boston$crime)
```

```
## [1] 0.8517787
```

```r
#KNN
set.seed(1)

train <- Boston
test <- Boston
train <- train[,c(2,3,5,6)]
test <- test[,c(2,3,5,6)]

crime <- Boston$crime
train <- as.matrix(na.omit(train))
test <- as.matrix(na.omit(test))

predknn <- knn(train,test,crime,k=1)
mean(predknn==Boston$crime)
```

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