Statistical Learning Homework 2

110024516 邱繼賢

```
library(mlbench)
library(corrplot)
library(dplyr)
library(ggplot2)
library(GGally)
library(caret)
library(ROSE)
library(MASS)
library(class)
library(psych)
library(nnet)
library(cowplot)
```

Problem 1.

EDA

觀察資料的數值特徵：

```
data("BreastCancer")
data1 = matrix(as.numeric(as.matrix(BreastCancer[,2:10])),699,9)
data1 = data.frame(data1)
colnames(data1) = colnames(BreastCancer[,2:10])
data1$case = as.factor(ifelse(BreastCancer$Class == "malignant",1,0))
data1 = data1[,c(10,1:9)]
dim(data1)
```

```
## [1] 699  10
```

```
summary(data1)
```

```
##  case     Cl.thickness      Cell.size        Cell.shape     Marg.adhesion
##  0:458   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1:241   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000
##          Median : 4.000   Median : 1.000   Median : 1.000   Median : 1.000
##          Mean   : 4.418   Mean   : 3.134   Mean   : 3.207   Mean   : 2.807
##          3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 4.000
##          Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##   Epith.c.size     Bare.nuclei      Bl.cromatin     Normal.nucleoli
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 2.000   Median : 1.000   Median : 3.000   Median : 1.000
##  Mean   : 3.216   Mean   : 3.545   Mean   : 3.438   Mean   : 2.867
##  3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                   NA's   :16
##     Mitoses
##  Min.   : 1.000
##  1st Qu.: 1.000
##  Median : 1.000
##  Mean   : 1.589
##  3rd Qu.: 1.000
##  Max.   :10.000
##
```

- 699 筆觀測值，10 個變數
- 其中 response variable *case* 為 2-levels 的 nomial variable：*case=1* 為惡性的類別
- 其餘 9 個 predictor variables 皆為 ordinal variables，數值落在 1~10 之間，代表著與乳癌有關的各種細胞或腫瘤數值
- 9 個 predictor variables 從數值的級距上看起來，分布皆有右偏趨勢，有很大一部分的資料都落在較小的數值

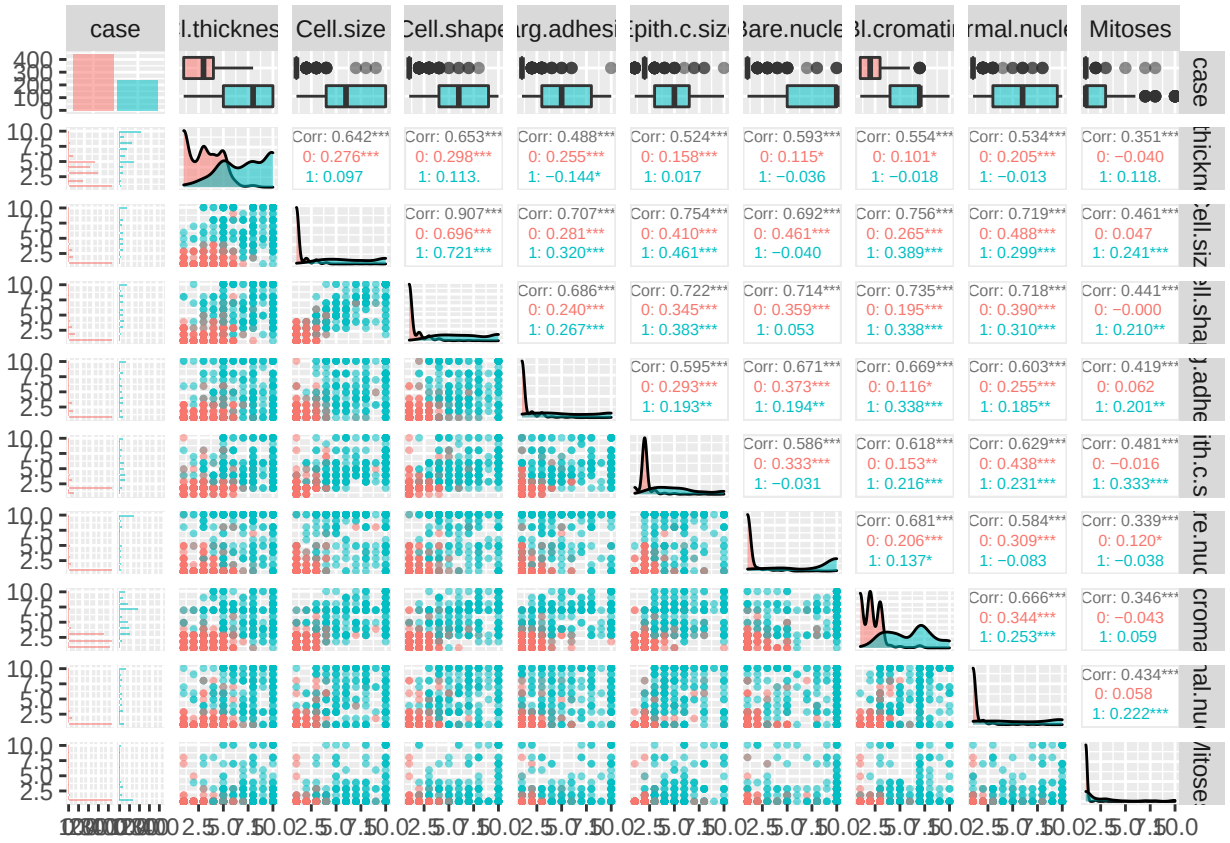
- 變數 *Bare.nuclei* 中有 16 個 NA 值，資料有所缺失，但只占全部資料中很小的比例，在後面的分析中將此 16 筆資料全部刪除

```
data1 = na.omit(data1)
dim(data1)
```

```
## [1] 683  10
```

將缺失值刪除後，剩下 683 筆觀測值，接下來對資料進行圖形上的分析：

```
ggpairs(data1, aes(color = data1$case, alpha= 0.5),
        upper = list(continuous = wrap("cor", size = 2)),
        lower = list(continuous = wrap("points", size=0.7)))
```



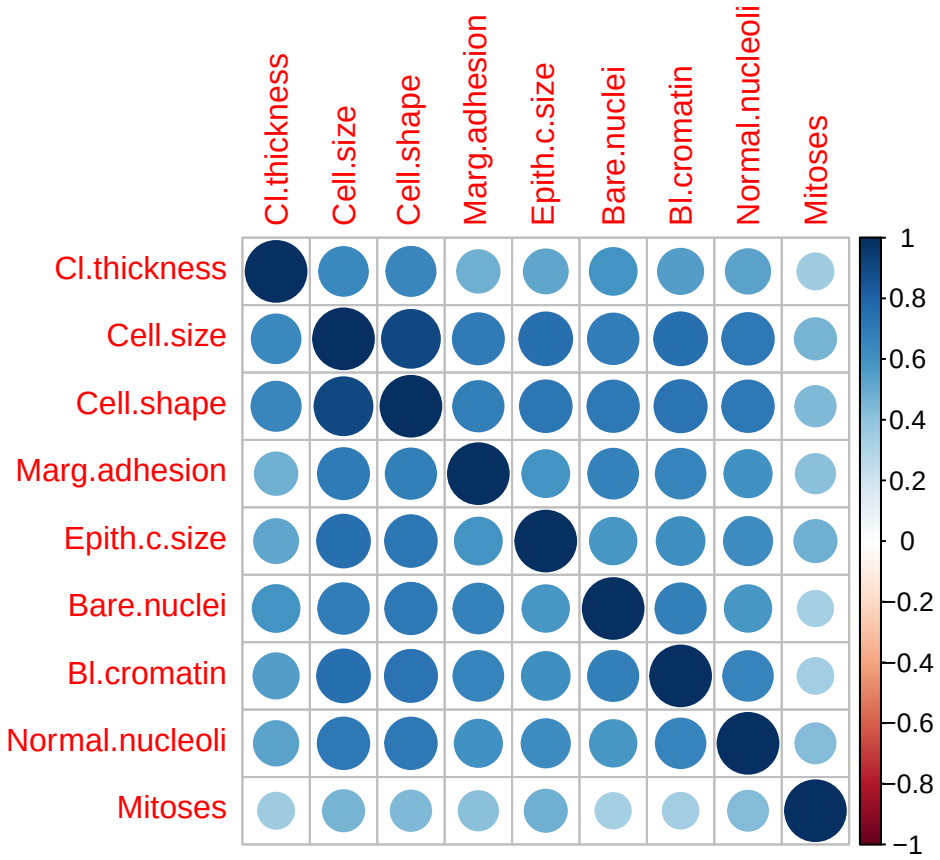
- 首先觀察 response variable *case* 對其餘變數的 side-by-side box-plots，可以發現被判定為良性 benign (*case=0* 紅色) 的病患在各項 predictor variables 的分數大多都集中在偏低的數值，和 malignant (*case=1* 藍色) 的病患分佈有明顯差距

- 也可以藉由對角線的 density plots 觀察到，紅色的資料幾乎都分布在較小的數值，這也是造成各 predictor variables 分布右偏的原因

- 再來觀察 pairwise scatter plots，發現紅色的資料點大多集中在左下角，代表著 benign 的病患在各 predictor variables 大多會同時呈現較小的數值

- correlation coefficient 的數值較多，不易觀察，以下直接使用 correlation plot 視覺化：

```
corrplot(cor(data1[,-1]))
```



- 9 個 predictor variables 之間都呈現為正相關，結合前面 predictor 和 response 之間的關係，可做出以下推論：此 9 個變數數值越大，則病患越有傾向具有惡性腫瘤，且這些變數間具有一定程度的正相關，背後可能有一個 latent variable 也就是「病患的身體狀況」

- *Cell.size* 和 *Cell.shape* 之間的正相關程度非常大，他們代表的意義分別為：「細胞大小的對稱性」和「細胞形狀的對稱性」，在建構模型時此兩變數可能會有共線性

接下來，將資料以 400:283 的比例隨機分成 training set 和 testing set，並利用 training set 來建構以下各種模型，然後觀察其在 testing set 上的表現：

Logistic regression

Model :

$$\log\left(\frac{p_{\text{case}}}{1-p_{\text{case}}}\right) = \beta_0 + \beta_1\ Cl.thickness + \beta_2\ Cell.size + \beta_3\ Cell.shape + \beta_4\ Marg.adhesion + \beta_5\ Epith.c.size$$
$$\beta_6\ Bare.nuclei + \beta_7\ Bl.cromatin + \beta_8\ Normal.nucleoli + \beta_9\ Mitoses$$

```
set.seed(10151)
idx = sample(1:683, 400, replace = F)
data1_train = data1[idx,] ; data1_test = data1[-idx,]
# logistic regression
glm.fit = glm(case ~ ., data1_train, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = case ~ ., family = binomial, data = data1_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.78221  -0.04868  -0.01641   0.00463   1.87892
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -14.24028    2.91657  -4.883 1.05e-06 ***
## Cl.thickness      0.81670    0.23209   3.519 0.000433 ***
## Cell.size         0.43729    0.40233   1.087 0.277077
## Cell.shape       -0.22373    0.42464  -0.527 0.598290
## Marg.adhesion     0.55602    0.18347   3.031 0.002441 **
## Epith.c.size     -0.05831    0.22961  -0.254 0.799531
## Bare.nuclei       0.35822    0.13688   2.617 0.008869 **
## Bl.cromatin       0.89900    0.27762   3.238 0.001203 **
## Normal.nucleoli   0.38128    0.18190   2.096 0.036077 *
## Mitoses           1.10696    0.51220   2.161 0.030683 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.708  on 399  degrees of freedom
## Residual deviance:  42.986  on 390  degrees of freedom
## AIC: 62.986
##
## Number of Fisher Scoring iterations: 9
```

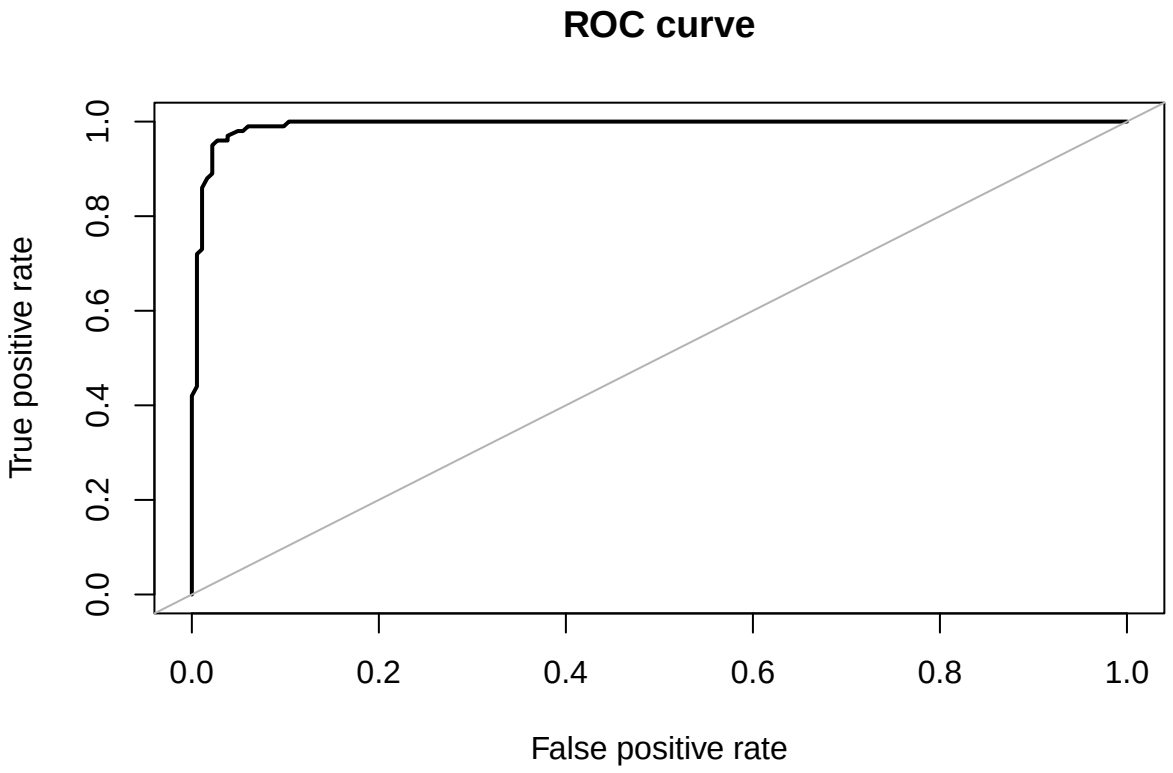
- 變數 *Cl.thickness, Marg.adhesion, Bare.nuclei, Bl.cromatin, Normal.nucleoli, Mitoses* 皆呈現顯著

```
glm.probs <- predict(glm.fit, data1_test, type = "response")
glm.pred = rep("benign (non-cased)", 283)
glm.pred[glm.probs > 0.5] = "malignant (cased)"
Direct = data1_test$case
levels(Direct) = c("benign (non-cased)", "malignant (cased)")
confusionMatrix(as.factor(glm.pred), Direct, positive = "malignant (cased)")
```

```
## Confusion Matrix and Statistics
##
##                     Reference
## Prediction           benign (non-cased) malignant (cased)
##   benign (non-cased)                179                10
##   malignant (cased)                   4                90
##
##                Accuracy : 0.9505
##                  95% CI : (0.9184, 0.9727)
##     No Information Rate : 0.6466
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8903
##
##  Mcnemar's Test P-Value : 0.1814
##
##             Sensitivity : 0.9000
##             Specificity : 0.9781
##          Pos Pred Value : 0.9574
##          Neg Pred Value : 0.9471
```

```
##              Prevalence : 0.3534
##          Detection Rate : 0.3180
##    Detection Prevalence : 0.3322
##       Balanced Accuracy : 0.9391
##
##        'Positive' Class : malignant (cased)
##
```

```
roc.curve(Direct, glm.probs, plotit = T)
```




```
## Area under the curve (AUC): 0.992
```

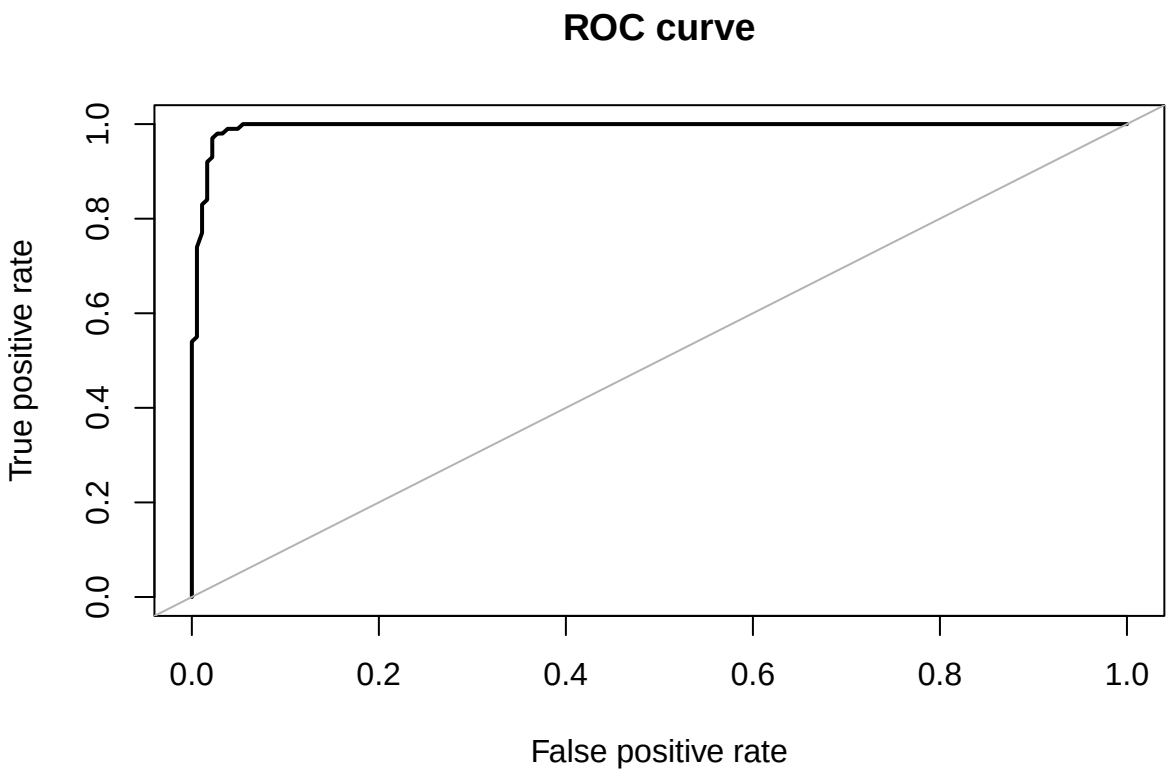
- 利用 confusion matrix 計算出 Accuracy = 0.9505
- ROC curve 表現相當好，AUC = 0.992

LDA

```
# LDA
lda.fit = lda(case ~ ., data1_train)
lda.pred = predict(lda.fit, data1_test)
lda.class = lda.pred$class
levels(lda.class) = c("benign (non-cased)", "malignant (cased)")
confusionMatrix(lda.class, Direct, positive = "malignant (cased)")
```

```
## Confusion Matrix and Statistics
##
##                     Reference
## Prediction           benign (non-cased) malignant (cased)
##   benign (non-cased)                179                  7
##   malignant (cased)                   4                 93
##
##                Accuracy : 0.9611
##                  95% CI : (0.9315, 0.9804)
##     No Information Rate : 0.6466
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9144
##
##  Mcnemar's Test P-Value : 0.5465
##
##             Sensitivity : 0.9300
##             Specificity : 0.9781
##          Pos Pred Value : 0.9588
##          Neg Pred Value : 0.9624
##              Prevalence : 0.3534
##          Detection Rate : 0.3286
##    Detection Prevalence : 0.3428
##       Balanced Accuracy : 0.9541
##
##        'Positive' Class : malignant (cased)
##
```

```
roc.curve(Direct, lda.pred$posterior[,2], plotit = T)
```




```
## Area under the curve (AUC): 0.994
```

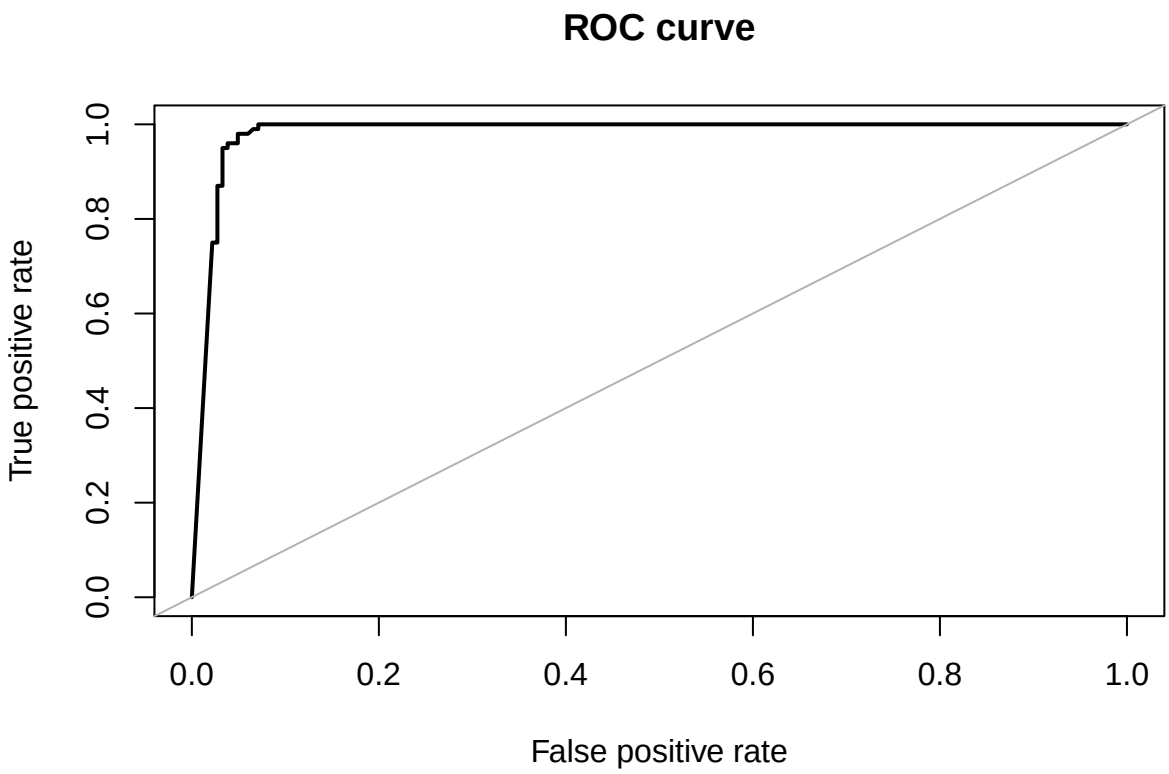
- 利用 confusion matrix 計算出 Accuracy = 0.9611
- ROC curve 表現跟 logistic regression 時差不多，AUC = 0.994

QDA

```
# QDA
qda.fit = qda(case ~ ., data1_train)
qda.pred = predict(qda.fit, data1_test)
qda.class = qda.pred$class
levels(qda.class) = c("benign (non-cased)", "malignant (cased)")
confusionMatrix(qda.class, Direct, positive = "malignant (cased)")
```

```
## Confusion Matrix and Statistics
##
##                     Reference
## Prediction           benign (non-cased) malignant (cased)
##   benign (non-cased)                174                 2
##   malignant (cased)                   9                98
##
##                Accuracy : 0.9611
##                  95% CI : (0.9315, 0.9804)
##     No Information Rate : 0.6466
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9163
##
##  Mcnemar's Test P-Value : 0.07044
##
##             Sensitivity : 0.9800
##             Specificity : 0.9508
##          Pos Pred Value : 0.9159
##          Neg Pred Value : 0.9886
##              Prevalence : 0.3534
##          Detection Rate : 0.3463
##    Detection Prevalence : 0.3781
##       Balanced Accuracy : 0.9654
##
##        'Positive' Class : malignant (cased)
##
```

```
roc.curve(Direct, qda.pred$posterior[,2], plotit = T)
```




```
## Area under the curve (AUC): 0.983
```

- 利用 confusion matrix 計算 accuracy = 0.9611
- ROC curve 表現也很好，AUC = 0.983

KNN

為了避免個變數因為單位不同而對資料點間距離造成影響，故在進行 KNN 之前先將資料對各變數 standardize

```
data1_std = scale(data1[,-1])
train_X = data1_std[idx,] ; train_Y = data1[idx,1]
test_X = data1_std[-idx,] ; test_Y = data1[-idx,1]
```

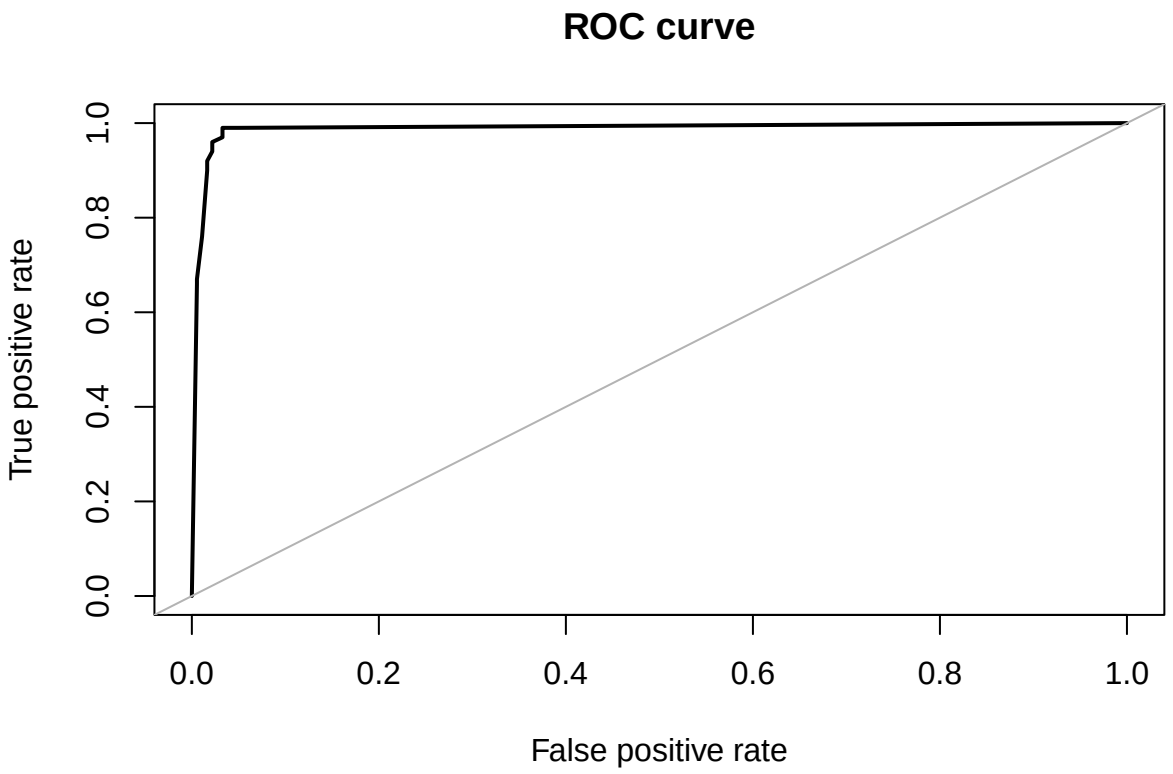
設定 k = 10 並對 testing data set 進行分類預測

```r
set.seed(1019)
knn_pred = knn(train_X, test_X, train_Y, k=10, prob = T)
levels(knn_pred) = c("benign (non-cased)", "malignant (cased)")
confusionMatrix(knn_pred, Direct, positive = "malignant (cased)")
```

```
## Confusion Matrix and Statistics
##
##                     Reference
## Prediction           benign (non-cased) malignant (cased)
##   benign (non-cased)                179                 5
##   malignant (cased)                   4                95
##
##                Accuracy : 0.9682
##                  95% CI : (0.9405, 0.9854)
##     No Information Rate : 0.6466
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9303
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9500
##             Specificity : 0.9781
##          Pos Pred Value : 0.9596
##          Neg Pred Value : 0.9728
##              Prevalence : 0.3534
##          Detection Rate : 0.3357
##    Detection Prevalence : 0.3498
##       Balanced Accuracy : 0.9641
##
##        'Positive' Class : malignant (cased)
##
```

```r
knn_prob = attributes(knn_pred)$prob
knn_prob = ifelse(knn_pred=="malignant (cased)"
                  ,knn_prob,1-knn_prob)
```

```
roc.curve(Direct, knn_prob, plotit = T)
```




```
## Area under the curve (AUC): 0.988
```

- 利用 confusion matrix 計算 accuracy = 0.9682
- ROC curve 表現很好，AUC = 0.988

Comparison and Conclusion

	Logistic	**LDA**	**QDA**	**KNN**
Accuracy	0.9505	0.9611	0.9611	0.9682
AUC	0.992	0.994	0.983	0.988

- 四種模型的 Accuracy 和 AUC 的表現差異不大，都非常好，這可能是因為 EDA 中有提到：*case=0* 和 *case=1* 的兩個類別資料分布的差異非常大

- 我們建構的模型只是對於此筆資料的表現很好，若用來預測未來 unknown observations 不見得還能有如此高的準確度

- 以上預測皆是在 Threshold = 0.5 的情況下做預測，解決實際問題時應考慮 False positive 和 False negative 時所需付出的成本差異來調整 Threshold

Problem 2.

EDA

觀察資料的各項數值特徵

```
data(Glass)
data2 = Glass
dim(data2)
```

```
## [1] 214  10
```

```
summary(data2)
```

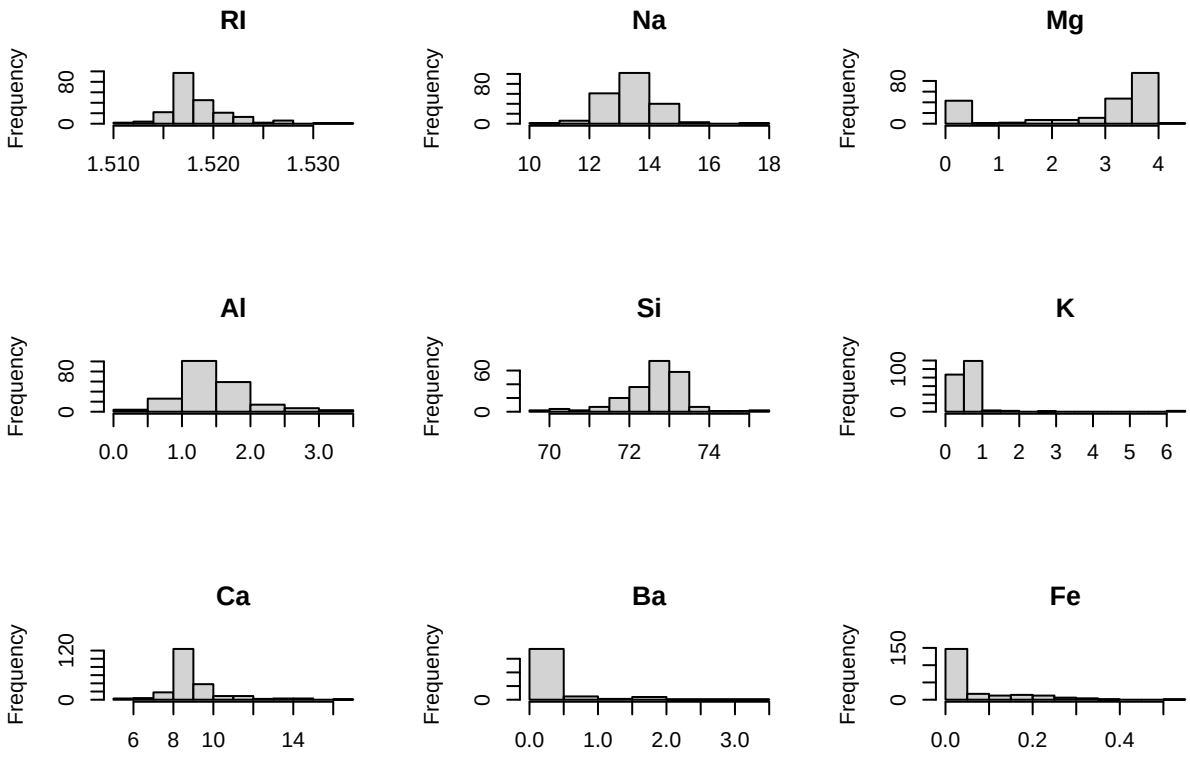
```
##        RI              Na              Mg              Al
##  Min.   :1.511   Min.   :10.73   Min.   :0.000   Min.   :0.290
##  1st Qu.:1.517   1st Qu.:12.91   1st Qu.:2.115   1st Qu.:1.190
##  Median :1.518   Median :13.30   Median :3.480   Median :1.360
##  Mean   :1.518   Mean   :13.41   Mean   :2.685   Mean   :1.445
##  3rd Qu.:1.519   3rd Qu.:13.82   3rd Qu.:3.600   3rd Qu.:1.630
##  Max.   :1.534   Max.   :17.38   Max.   :4.490   Max.   :3.500
##        Si              K               Ca              Ba
##  Min.   :69.81   Min.   :0.0000   Min.   : 5.430   Min.   :0.000
##  1st Qu.:72.28   1st Qu.:0.1225   1st Qu.: 8.240   1st Qu.:0.000
##  Median :72.79   Median :0.5550   Median : 8.600   Median :0.000
##  Mean   :72.65   Mean   :0.4971   Mean   : 8.957   Mean   :0.175
##  3rd Qu.:73.09   3rd Qu.:0.6100   3rd Qu.: 9.172   3rd Qu.:0.000
##  Max.   :75.41   Max.   :6.2100   Max.   :16.190   Max.   :3.150
##        Fe          Type
##  Min.   :0.00000   1:70
##  1st Qu.:0.00000   2:76
```

```
##  Median :0.00000   3:17
##  Mean   :0.05701   5:13
##  3rd Qu.:0.10000   6: 9
##  Max.   :0.51000   7:29
```

- 214 筆觀測值，10 個變數
- 其中 response variable *Type* 為類別型變數，共 7 個 levels，代表七種不同類型的玻璃，但此筆資料中並未出現 *Type*=4 的種類，大部分的資料都是 *Type*=1,2,7 這三種類別
- 其餘 9 個變數皆為 predictor variables，*RI*(refractive index) 為折射率，其他 8 個變數則代表玻璃中該金屬元素的含量
- 變數 *Ba, Fe* 有超過一半的資料點數值為零

觀察 9 個 predictor variables 的 histogram

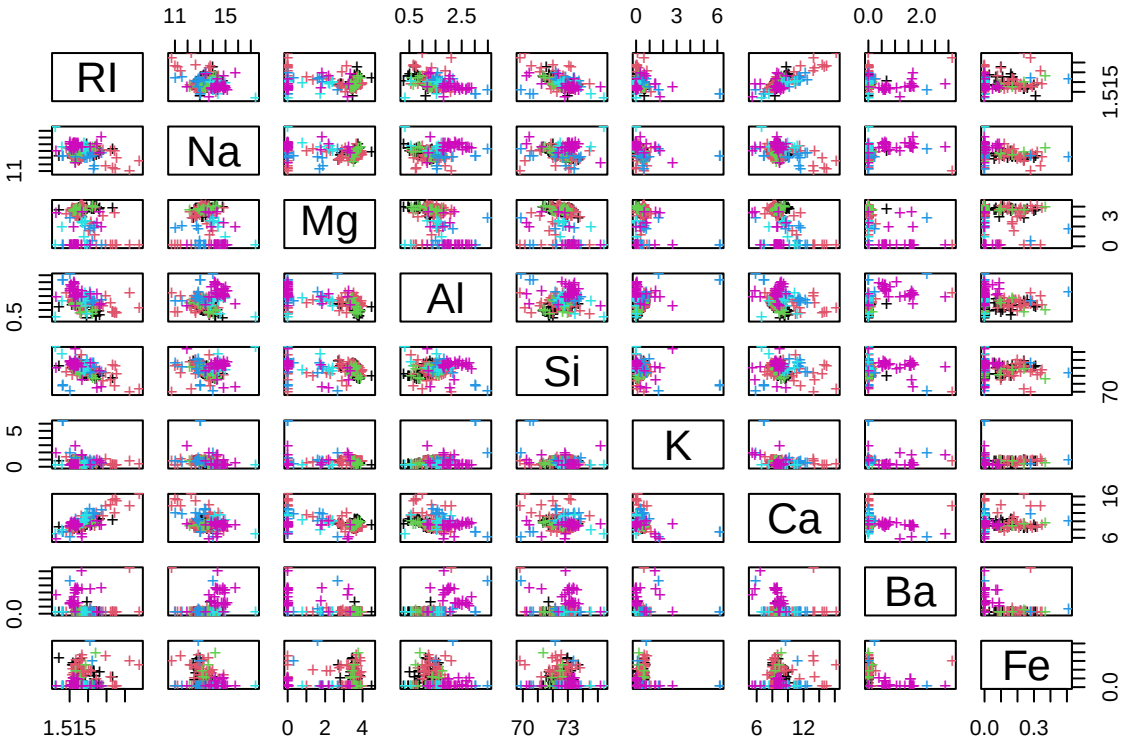
```
par(mfrow = c(3,3))
for (i in 1:9) {
  hist(data2[,i], xlab = "", main = names(data2)[i])
}
```



- *Mg* 有著明顯的雙峰分布，可能代表著不同的玻璃類別
- *Ba, Fe* 大部分的資料點都為零，分布呈現明顯右偏

觀察變數間的 pairwise scatter plot

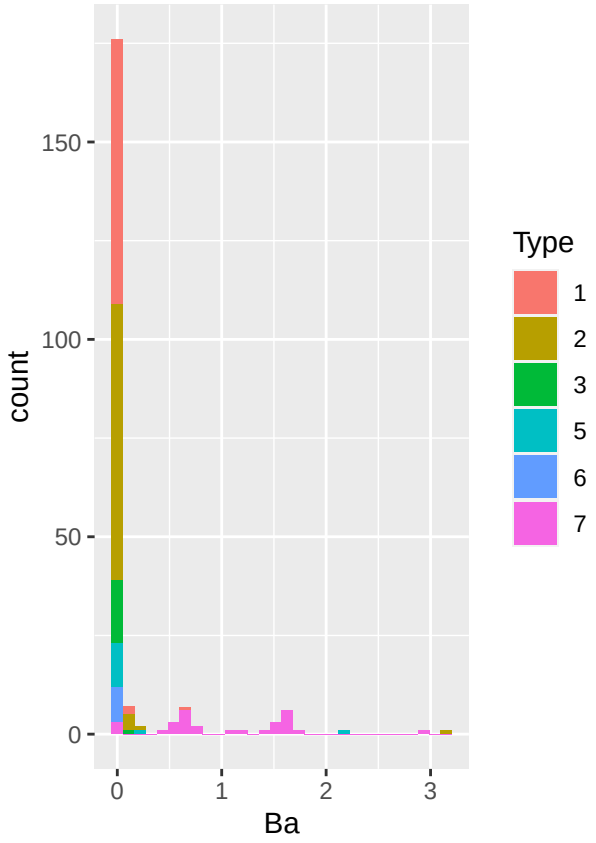
```
pairs(data2[,-10], col=data2[,10], pch="+")
```

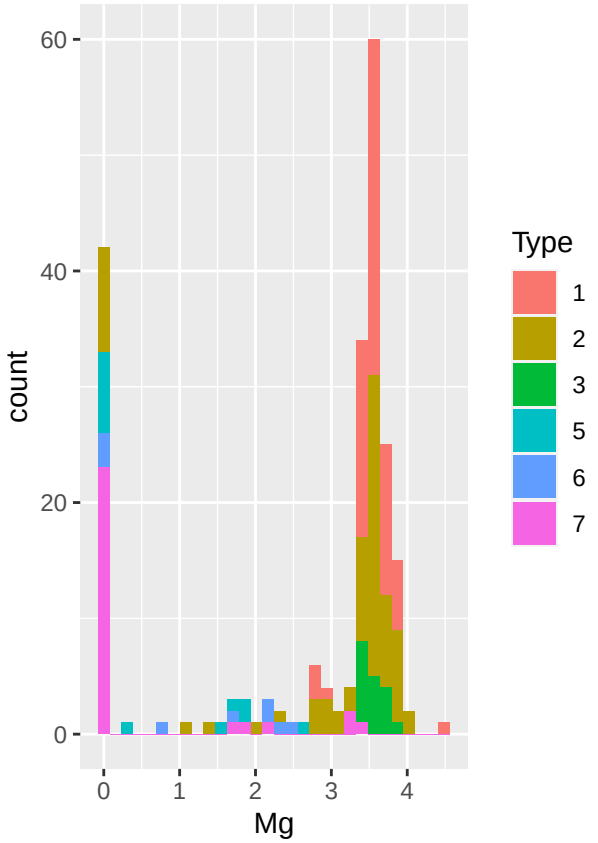


- 變數 RI 和 Ca 之間具有明顯的正向線性相關，可以推論出 Ca 元素的多寡可能會線性的影響折射率的大小
- 變數 Ba>0 的資料點大多為粉色 ($Type$=7) 的資料
- 其餘變數看不出太明顯的關係

將 Ba 和 Mg 的 histogram 根據不同的 $Type$ 作圖

```
p1 = ggplot(data2, aes(Ba, fill = Type)) +
  geom_histogram()
p2 = ggplot(data2, aes(Mg, fill = Type)) +
  geom_histogram()
plot_grid(p1,p2)
```



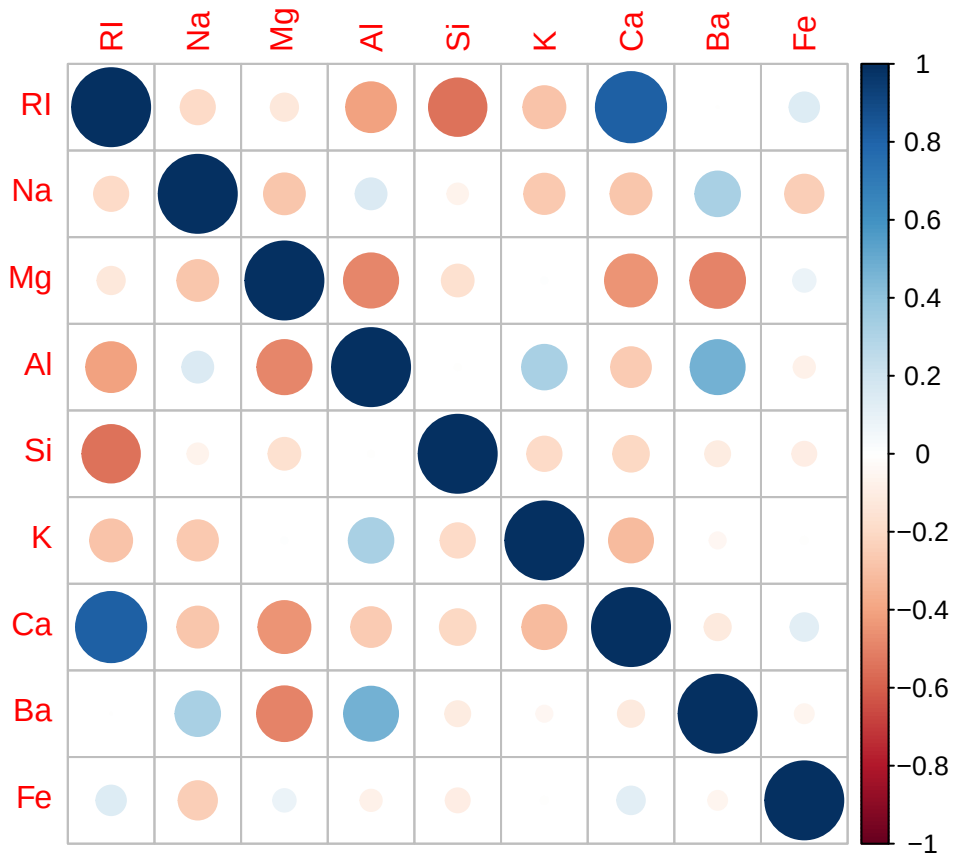


- 除了 *Type*=7 的類別其對應到的 *Ba* 為正，其餘類別大多 *Ba*=0，故玻璃內是否含有 *Ba* 元素可能為分辨是否為 *Type*=7 的重要變數

- *Type*=7 類別對應到的 *Mg* 大多為零，而 *Type*=1 則大多介於 3~4 之間較大的數值，故玻璃內是否含有 *Mg* 元素可能為區分 *Type*=1 or 7 的重要變數

觀察 9 個 predictor variables 之間的 correlation plot

```
corrplot(cor(Glass[,-10]))
```



- *RI* 和 *Ca* 呈現高度正相關，這與前面 pairwise scatter plot 所做出的結論一致

- *RI* 和 *Si* 呈現中度負相關，可推測玻璃內 *Si* 元素的增加會降低其折射率

將資料以 150:64 的比例隨機分配成 training data 和 testing data，並在 training data 上用以下各種分類方式建構模型，然後對 testing data 進行預測

Multiclass logistic regression (multinomial regression)

Model :

$$P\left(Type = k\right) \ = \ \frac{\exp(X\beta_k)}{\sum_{l=1}^{7}\exp(X\beta_l)} \ , \ k = 1, ..., 7$$

where X is the model matrix and β_l is a vector with length $= 10$

```
set.seed(1020)
idx = sample(1:214, 150)
data2_train = data2[idx,]
data2_test = data2[-idx,]
```

```
Direct = data2_test$Type
fit_mul = multinom(Type ~ ., data = data2_train)
```

```
## # weights:  66 (50 variable)
## initial  value 268.763920
## iter  10 value 165.612049
## iter  20 value 116.106759
## iter  30 value 107.302289
## iter  40 value 104.105738
## iter  50 value 101.704752
## iter  60 value 100.559070
## iter  70 value 100.140600
## iter  80 value 99.963806
## iter  90 value 99.578987
## iter 100 value 98.863872
## final  value 98.863872
## stopped after 100 iterations
```

```
pred.prob_mul = predict(fit_mul, data2_test, type = "probs")
pred.class_mul = predict(fit_mul, data2_test, type = "class")
```

```
table(pred.class_mul, Direct)
```

```
##                Direct
## pred.class_mul  1  2  3  5  6  7
##              1 14  8  4  0  0  0
##              2  5 15  3  0  0  0
##              3  0  0  0  0  0  0
##              5  0  1  0  1  0  5
##              6  0  0  0  0  2  0
##              7  0  1  0  0  0  5
```

```
mean(pred.class_mul == Direct)
```

```
## [1] 0.578125
```

- Accuracy = 0.5781

LDA

```
fit_lda = lda(Type ~ ., data = data2_train)
pred.class_lda = predict(fit_lda, data2_test)$class
table(pred.class_lda, Direct)
```

```
##                Direct
## pred.class_lda  1  2  3  5  6  7
##              1 15  7  3  0  0  0
##              2  4 16  4  0  1  0
##              3  0  0  0  0  0  0
##              5  0  2  0  1  0  0
##              6  0  0  0  0  1  0
##              7  0  0  0  0  0 10
```

```
mean(pred.class_lda == Direct)
```

```
## [1] 0.671875
```

- Accuracy = 0.6719 明顯高於 multinomial regression

KNN

先對各 predictor variables 做 standardize 然後選定 k=3 進行 KNN 預測

```
data2_std = scale(data2[,-10])
train_X = data2_std[idx,] ; train_Y = data2[idx,10]
test_X = data2_std[-idx,]
set.seed(1020)
pred.class_knn = knn(train_X, test_X, train_Y, k = 3)
table(pred.class_knn, Direct)
```

```
##                Direct
## pred.class_knn  1  2  3  5  6  7
##              1 17  5  4  0  0  1
##              2  2 20  2  0  0  0
##              3  0  0  1  0  0  0
```

```
##             5  0  0  0  1  0  1
##             6  0  0  0  0  2  0
##             7  0  0  0  0  0  8
```

```
mean(pred.class_knn == Direct)
```

```
## [1] 0.765625
```

- Accuracy = 0.7656 明顯又高於 LDA

Comparison and Conclusion

	Multinomial	**LDA**	**KNN**
Accuracy	0.5781	0.6719	0.7656

- KNN 的表現最好，而 Multinomial regression 分類表現最差

- 以上的分類結果皆是考慮各類別的預測機率最大者則分為該類，並未考慮各類別分類錯誤時可能有著不同的成本

- 此筆資料數總共僅只有 214 筆，且有部分類別的個數相當少，在此情況下所分割出的 training data 和 testing data 可能無法代表整個母體