# 統計學習 作業四

106070038 科管院學士班 杜葳葳

10.

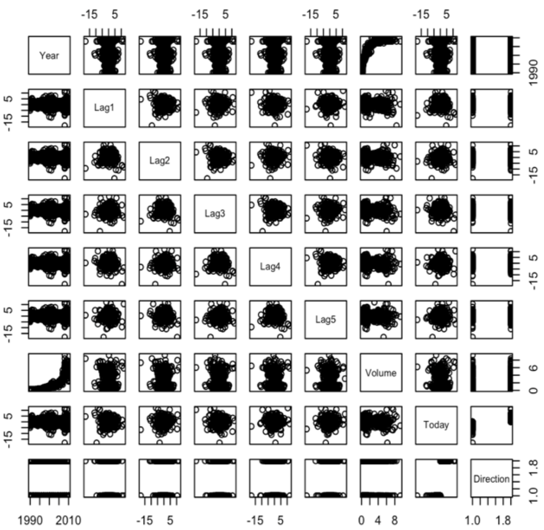
(a) Weekly 是一組**二維(1089*9)** S&P500 股票市場自 1990-2010 年的資料

1089 個 row：21 年*每年約 51.85 週

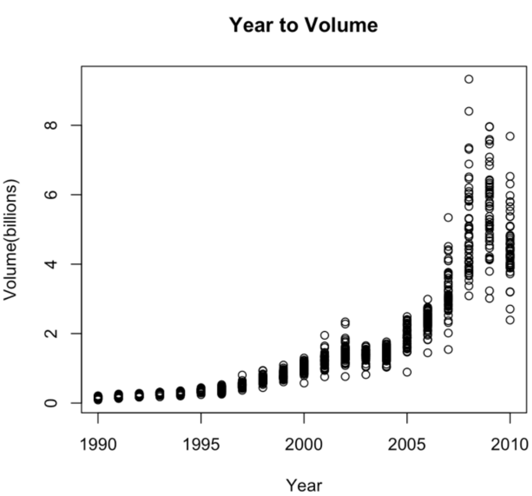
9 個 column：年份、上週報酬率、兩週前報酬率、三週前報酬率、四週前報酬率、五週前報酬率、交易量、本週報酬率、未來走勢(上或下)

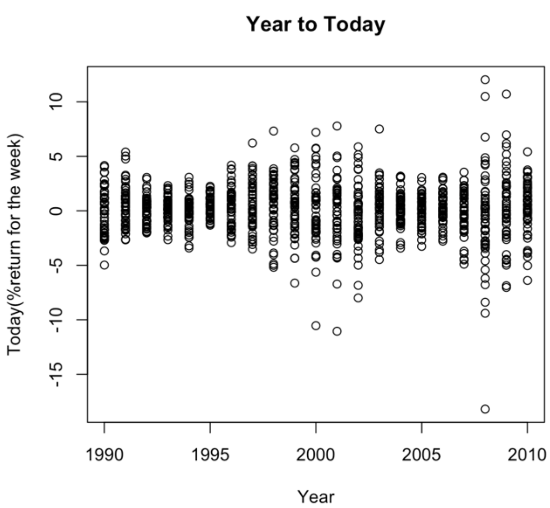
- Today、Lag1 到 Lag5 的最大值、最小值、四分位數、平均值、中位數大致都相同，因為第一週的 Today＝第二週的 Lag1=第三週的 Lag2=第四週的 Lag3=第五週的 Lag4=第六週的 Lag5…以此類推，但因為頭尾的資料(1990 前五週和 2010 後五週)沒有重複，所以最大值、最小值、四分位數、平均值、中位數在 Today、Lag1 到 Lag5 有些許差異。
- 將所有資料的 Direction 做分析，Down 出現 484 次、Up 出現 605 次，可知在這 21 年之中有 55.555％的機率股價是向上的。
- 用 pairs()繪製兩兩成對之散佈圖矩陣，可觀察到 volume 和 year，有正相關的趨勢

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




- 用 Year 和 Volume 做圖，可觀察到從 1990 至 2010 年，隨著年份增加交易量(Volume)也跟著增加。
- 用 Year 和 Today 做圖，可觀察到在 2000 年前後和 2008 年前後，週報酬率波動較為劇烈。



(b) 只有 **Lag2** 呈現統計顯著，估計係數為 0.05844，Lag2 上升一單位、Direction 的 odds 上升 e^0.058 單位。

```
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) 混淆矩陣如下：

```
pred    Down  Up
  Down    54  48
  Up     430 557
```

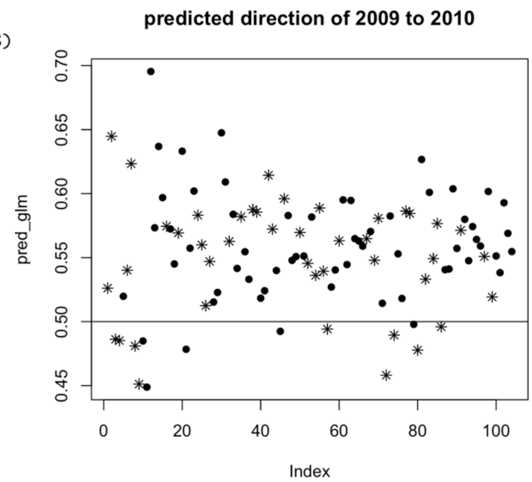
True Positive: 557、False Positive: 430、True Negative: 54、False Negative: 48

Accuracy: 0.5610652
Recall: 0.92066116
Precision: 0.56433637

根據 Precision 可知，判斷為 Up 的情況下，僅 56.433637%真的為 Up，儘管 True Negative 很高，但 False Negative 亦很高，**False Negative** 的情況造成 model 很大的**偏誤**。

(d) 拿 1990-2008 年的資料當 training data，用 Lag2 做 logistic regression model 來預測 Direction，由下圖可知，**Lag2** 呈現統計顯著，估計係數為 0.05810。
接著用此模型 predict 2009-2010 (testing data)，**準確率(Accuracy)為 0.625**。(混淆矩陣在下頁最上方)
右下散佈圖為用 2009-2010 年的 Direction 做圖，黑線以上(pred_glm>0.5)為模型預測為 Up 的資料、黑線以下則為模型預測為 Down 的資料。另外，圓點為實際上是 Up 的資料、星點為實際上是 Down 的資料。

```
Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly_1990_2008)

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
```



```
      Direction_2009_2010
pred2  Down Up
  Down    9  5
  Up     34 56
```

(e) 用 LDA 重複(d)小題，拿 1990-2008 年的資料當 training data，用 Lag2 來預測 Direction，由下圖可知，**Lag2 並沒有**呈現統計顯著。

接著用此模型 predict 2009-2010 (testing data)，混淆矩陣如下圖，**準確率(Accuracy)為 0.625**，和(d)小題相同。

```
Call:
lda(Direction ~ Lag2, data = Weekly_1990_2008, family = binomial)

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
```

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

(f) 用 QDA 重複(d)小題，拿 1990-2008 年的資料當 training data，用 Lag2 來預測 Direction，由下圖可知，**Lag2 並沒有**呈現統計顯著。

接著用此模型 predict 2009-2010 (testing data)，混淆矩陣如下圖，**準確率(Accuracy)為 0.58653846**，是三個模型中最差的。

```
Call:
qda(Direction ~ Lag2, data = Weekly_1990_2008, family = binomial)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

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

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

(g) 用 KNN 重複(d)小題(**K=1**)，拿 1990-2008 年的資料當 training data，用 Lag2 來預測 Direction，接著用此模型預測 2009-2010 (testing data)的 Direction，混淆矩陣如下圖，**準確率(Accuracy)為 0.5**，是以上四個模型(含)中最差的。

```
pred_knn_1 Down Up
         1   21 30
         2   22 31
```

(g-2) 用 Naïve Bayes 重複(d)小題，拿 1990-2008 年的資料當 training data，用 Lag2 來預測 Direction，接著用此模型預測 2009-2010 (testing data)的 Direction，混淆矩陣在下頁最上方，**準確率(Accuracy)為 0.5865385**。

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
     Down        Up 
0.4477157 0.5522843 

Conditional probabilities:
      Lag2
Y             [,1]     [,2]
  Down -0.03568254 2.199504
  Up    0.26036581 2.317485
```

```
pred_nb Down Up
   Down    0  0
   Up     43 61
```

(h) 以上五個模型的 Accuracy 相比，依序為：**Logistic Regression=LDA>Naïve Bayes>QDA>KNN**
Logistic Regression 和 LDA 提供最好的結果。

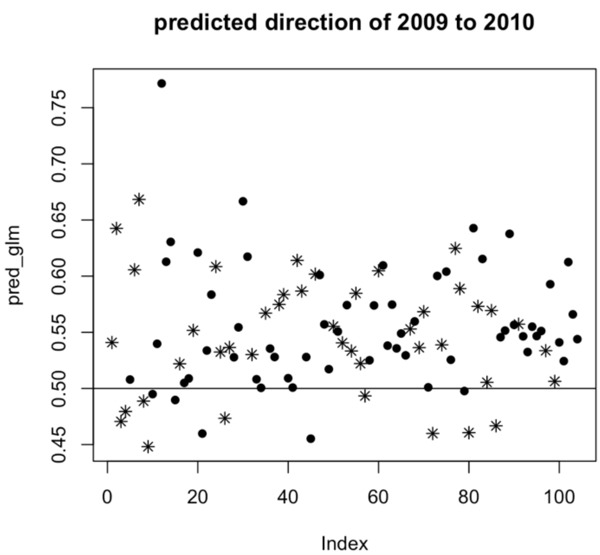
(i)

**Logistic Regression**：

多次嘗試各種參數組合，Accuracy 皆無法比單純使用 Lag2(d 小題)來得好，然而，若使用 Lag2 到 Lag5 為參數，混淆矩陣與 Lag2(d 小題)相同。

右下散佈圖為用 2009-2010 年的 Direction 做圖，黑線以上(pred_glm>0.5)為模型預測為 Up 的資料、黑線以下則為模型預測為 Down 的資料。另外，圓點為實際上是 Up 的資料、星點為實際上是 Down 的資料。

```
Call:
glm(formula = Direction ~ Lag2 + Lag3 + Lag4 + Lag5, family = binomial, 
    data = Weekly_1990_2008)

Deviance Residuals: 
   Min      1Q  Median      3Q     Max  
-1.781  -1.254   1.005   1.093   1.394  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  0.21193    0.06475   3.273  0.00106 **
Lag2         0.05710    0.02901   1.969  0.04901 * 
Lag3        -0.01232    0.02900  -0.425  0.67110   
Lag4        -0.02117    0.02883  -0.734  0.46291   
Lag5        -0.03077    0.02887  -1.066  0.28647   
---
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: 1348.8  on 980  degrees of freedom
AIC: 1358.8

Number of Fisher Scoring iterations: 4
```




```
       Direction_2009_2010
pred2   Down Up
   Down    9  5
   Up     34 56
```

**LDA**：

多次嘗試各種參數組合，Accuracy 皆無法比單純使用 Lag2(e 小題)來得好，然而，若使用 Lag2 和 Lag3 為參數，Accuracy 與 Lag2(e 小題)相同。

```
Call:
lda(Direction ~ Lag2 + Lag3, data = Weekly_1990_2008, family = binomial)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2       Lag3
Down -0.03568254 0.17080045
Up    0.26036581 0.08404044

Coefficients of linear discriminants:
             LD1
Lag2  0.42459797
Lag3 -0.08880475
```

|  | Down | Up |
|---|---|---|
| Down | 8 | 4 |
| Up | 35 | 57 |

**QDA：**

多次嘗試後發現，若使用 Lag2+Lag3 為參數，Accuracy 為 0.60576923，比(f)小題還高。

```
Call:
qda(Direction ~ Lag2 + Lag3, data = Weekly_1990_2008, family = binomial)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2       Lag3
Down -0.03568254 0.17080045
Up    0.26036581 0.08404044
```

|  | Down | Up |
|---|---|---|
| Down | 4 | 2 |
| Up | 39 | 59 |

**KNN**：當 **K=4** 時，**Accuracy=0.61538462**，比(g) K=1 的 model 還準確。

| pred_knn_1 | Down | Up |
|---|---|---|
| 1 | 20 | 17 |
| 2 | 23 | 44 |

**Naïve Bayes**：多次嘗試各種參數組合，皆無法比單純使用 Lag2(g-2 小題) 的 Accuracy 來得高，然而，若使用 Lag3 和 Lag4 為參數，Accuracy 與 Lag2(e 小題)相同。

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
     Down        Up 
0.4477157 0.5522843 

Conditional probabilities:
      Lag3
Y            [,1]     [,2]
  Down 0.17080045 2.228462
  Up   0.08404044 2.309105

      Lag4
Y            [,1]     [,2]
  Down 0.15925624 2.400042
  Up   0.09220956 2.165612
```

| pred_nb | Down | Up |
|---|---|---|
| Down | 9 | 9 |
| Up | 34 | 52 |

```
> mean(pred_nb == Weekly_2009_2010$Direction)
[1] 0.5865385
```

總結以上，同(h)小題的結果，Logistic Regression 和 LDA 提供最好的正確率。

11.

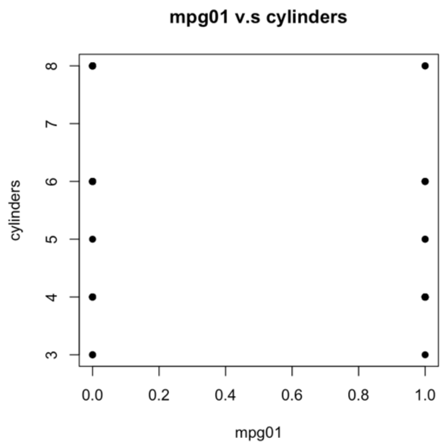
(a) mpg 的中位數為 **22.75**，先將 mpg01 的所有欄位設為 0，再判斷 mpg>0.5 者的 mpg01 改為 1，mpg01 的最大值、最小值、四分位數、平均數、中位數如右下所示。

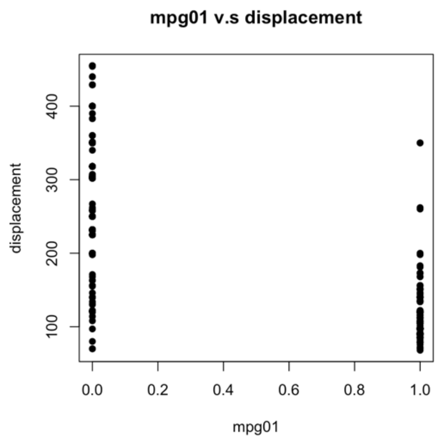
```
  mpg cylinders displacement horsepower weight acceleration
1  18         8          307        130   3504         12.0
2  15         8          350        165   3693         11.5
3  18         8          318        150   3436         11.0
4  16         8          304        150   3433         12.0
5  17         8          302        140   3449         10.5
6  15         8          429        198   4341         10.0
  year origin                      name mpg01
1   70      1 chevrolet chevelle malibu     0
2   70      1         buick skylark 320     0
3   70      1        plymouth satellite     0
4   70      1             amc rebel sst     0
5   70      1               ford torino     0
6   70      1          ford galaxie 500     0
```

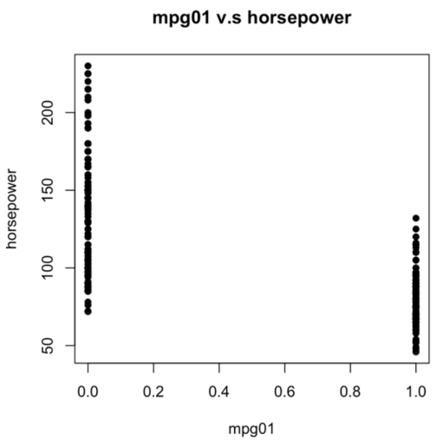
```
summary(Auto$mpg01)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.0     0.0     0.5     0.5     1.0     1.0 
```

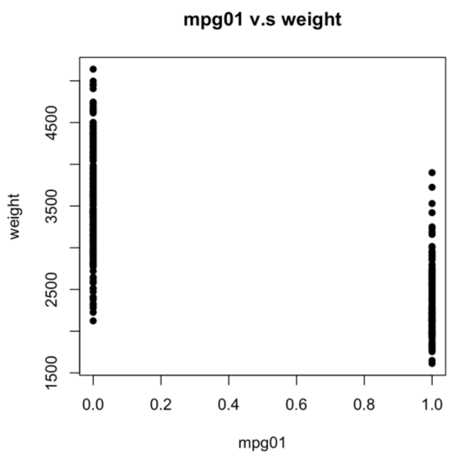
(b) 計算所有因子和 mpg01 的 correlation，可觀察到 mpg01 與 mpg 為高度正相關、與 origin/year/acceleration 為中度正相關、與 horsepower 為中度負相關、與 cylinders/displacement/weight 為高度負相關。各個因子與 mpg01 的作圖如下，從圖中難以看出關聯性，故建模時會直接用 correlation 的絕對值來判斷要選擇什麼因子。

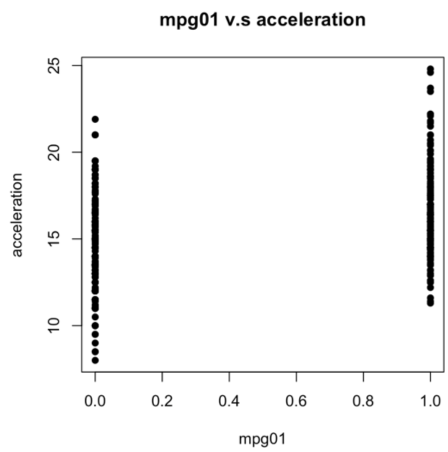
```
> cor(Auto$mpg01,na.omit(Auto[-9]))
           mpg  cylinders displacement horsepower     weight acceleration      year    origin mpg01
[1,] 0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215 0.4299042 0.5136984     1
```

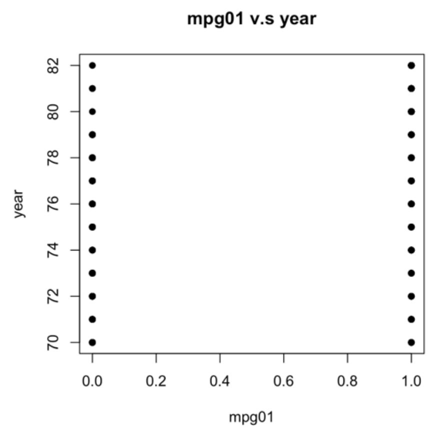


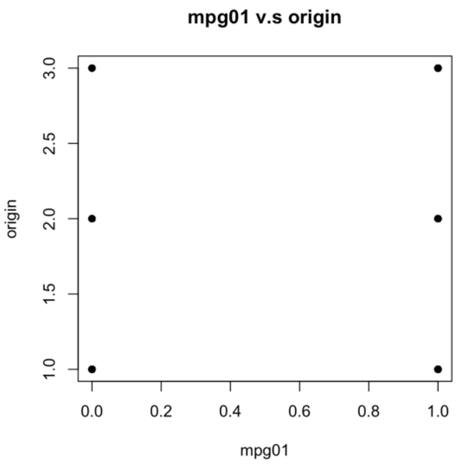






(c) 將資料切割成 training set 和 testing set，使用 training：testing = 0.75：0.25 的比例

(d) LDA

若使用單一因子，**cylinders** 和 mpg01 的 correlation 絕對值最高的，訓練出來的模型 test error = **0.06122449** 最低 (如下圖二的方框所示)

另外，如果使用所有因子，test error = 0.07142857 (如下圖五的方框所示)，並沒有比只用 cylinders 單一因子的 test error 低。

```
Call:
lda(mpg01 ~ origin, data = training, family = binomial)

Prior probabilities of groups:
        0         1 
0.5068027 0.4931973 

Group means:
    origin
0 1.167785
1 2.000000

Coefficients of linear discriminants:
            LD1
origin 1.440064
> lda.pred = predict(auto_lda,testing)
> table(lda.pred$class,testing$mpg01)
   
     0  1
  0 42 19
  1  5 32
```

```
Call:
lda(mpg01 ~ cylinders, data = training, family = binomial)

Prior probabilities of groups:
        0         1 
0.5068027 0.4931973 

Group means:
  cylinders
0  6.744966
1  4.193103

Coefficients of linear discriminants:
                LD1
cylinders 0.8665057
> lda.pred = predict(auto_lda,testing)
> table(lda.pred$class,testing$mpg01)
   
     0  1
  0 43  2
  1  4 49
```

```
Call:
lda(mpg01 ~ origin + year + acceleration, data = training, family = binomial)

Prior probabilities of groups:
        0         1 
0.5068027 0.4931973 

Group means:
    origin     year acceleration
0 1.167785 74.49664     14.59060
1 2.000000 77.39310     16.46552

Coefficients of linear discriminants:
                   LD1
origin       1.0809486
year         0.1534729
acceleration 0.1578155
> lda.pred = predict(auto_lda,testing)
> table(lda.pred$class,testing$mpg01)
   
     0  1
  0 39 15
  1  8 36
```

```
Call:
lda(mpg01 ~ displacement + origin + year + acceleration, data = training, 
    family = binomial)

Prior probabilities of groups:
        0         1 
0.5068027 0.4931973 

Group means:
  displacement   origin     year acceleration
0     270.5168 1.167785 74.49664     14.59060
1     115.9138 2.000000 77.39310     16.46552

Coefficients of linear discriminants:
                     LD1
displacement -0.01333499
origin        0.20008164
year          0.09787900
acceleration -0.04675247
> lda.pred = predict(auto_lda,testing)
> table(lda.pred$class,testing$mpg01)
   
     0  1
  0 41  2
  1  6 49
```

```
Call:
lda(mpg01 ~ cylinders + horsepower + weight + displacement + 
    origin + year + acceleration, data = training, family = binomial)

Prior probabilities of groups:
        0         1 
0.5068027 0.4931973 

Group means:
  cylinders horsepower   weight displacement   origin     year acceleration
0  6.744966  130.14094 3618.268     270.5168 1.167785 74.49664     14.59060
1  4.193103   79.26897 2326.607     115.9138 2.000000 77.39310     16.46552

Coefficients of linear discriminants:
                      LD1
cylinders    -0.387638973
horsepower    0.009630520
weight       -0.001298945
displacement  0.001197267
origin        0.237130689
year          0.115165258
acceleration  0.041016899
> lda.pred = predict(auto_lda,testing)
> table(lda.pred$class,testing$mpg01)
   
     0  1
  0 42  2
  1  5 49
```

(e) QDA

對除了 name 以外的所有 column 計算 correlation，方便後續挑選模型因子。

```
> cor(na.omit(Auto[-9]),na.omit(Auto[-9]))
                    mpg  cylinders displacement horsepower     weight acceleration       year     origin      mpg01
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285  0.5805410  0.5652088  0.8369392
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -0.3456474 -0.5689316 -0.7591939
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.6145351 -0.7534766
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.4551715 -0.6670526
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.5850054 -0.7577566
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.2127458  0.3468215
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000  0.1815277  0.4299042
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458  0.1815277  1.0000000  0.5136984
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.5136984  1.0000000
```

如(d)小題的結果，單一因子 test error 最小的為和 mpg01 的 correlation 絕對值最高的 **cylinders**，**test error = 0.06122449** (混淆矩陣如下圖一方框所示)。

若使用所有因子的 error rate = 0.07142857(混淆矩陣如下圖二方框所示)。

另外，有嘗試挑選 cylinders 和其他與 cylinders correlation 較低的因子，但 error rate 與使用所有因子相同，但混淆矩陣有些許差異，若較重視 true positive，則可選擇此模型(混淆矩陣如下圖三方框所示)。

```
Call:
qda(mpg01 ~ cylinders, data = training, family = binomial)

Prior probabilities of groups:
        0         1
0.5068027 0.4931973

Group means:
  cylinders
0  6.744966
1  4.193103
> qda.pred = predict(auto_qda,testing)
> table(qda.pred$class, testing$mpg01)

    0  1
0  43  2
1   4 49
```

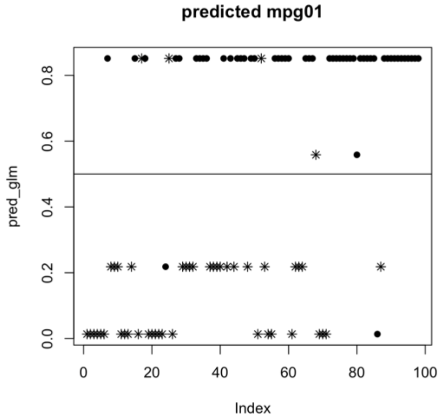
```
Call:
qda(mpg01 ~ cylinders + displacement + horsepower + weight +
    acceleration + year + origin, data = training, family = binomial)

Prior probabilities of groups:
        0         1
0.5068027 0.4931973

Group means:
  cylinders displacement horsepower   weight acceleration     year   origin
0  6.744966     270.5168  130.14094 3618.268     14.59060 74.49664 1.167785
1  4.193103     115.9138   79.26897 2326.607     16.46552 77.39310 2.000000
> qda.pred = predict(auto_qda,testing)
> table(qda.pred$class, testing$mpg01)

    0  1
0  44  4
1   3 47
```

```
Call:
qda(mpg01 ~ cylinders + year + acceleration + origin, data = training,
    family = binomial)

Prior probabilities of groups:
        0         1
0.5068027 0.4931973

Group means:
  cylinders     year acceleration   origin
0  6.744966 74.49664     14.59060 1.167785
1  4.193103 77.39310     16.46552 2.000000
> qda.pred = predict(auto_qda,testing)
> table(qda.pred$class, testing$mpg01)

    0  1
0  42  2
1   5 49
```

(f) Logistic Regression

選擇和 mpg01 的 correlation 的絕對值最高的 **cylinders** 做 logistic regression 來預測 mpg01，cylinders 有統計顯著性，testing data 的混淆矩陣如下，**test error = 0.06122449**。另外，將 testing data 的結果會製成散佈圖，黑線以上為預測 mpg01 為 1 的、黑線以下為預測 mpg01 為 0 的，黑點為實際上 mpg01 為 1 的、星狀點則為實際上 mpg01 為 0 的。

```
Call:
glm(formula = mpg01 ~ cylinders, family = binomial, data = training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5667  -0.1643  -0.1643   0.5673   2.9365

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   7.7888     0.7999   9.737   <2e-16 ***
cylinders    -1.5108     0.1614  -9.359   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 407.52  on 293  degrees of freedom
Residual deviance: 203.35  on 292  degrees of freedom
AIC: 207.35

Number of Fisher Scoring iterations: 6
```




```
pred_mpg01  0  1
         0 43  2
         1  4 49
```

(g) KNN

用 **cylinders** 來建 KNN 模型預測 mpg01 的 **error rate=0.06122449(K=1)**是最小的，且隨著 K 增加、error rate 皆相同。

```
auto_pred_knn_1  0  1
              0 43  2
              1  4 49
```

附錄：R 程式碼

```
# 10.
install.packages("stats")
library(stats)
install.packages("MLmetrics")
library(MLmetrics)
install.packages("e1071")
library(e1071)
install.packages("ISLR")
library(ISLR)
install.packages("MASS")
library(MASS)
install.packages("class")
library(class)
install.packages("tidyverse")
library(tidyverse)
data(Weekly)
head(Weekly,10)

#(a)
?Weekly
summary(Weekly)
dim(Weekly)
str(Weekly)
plot(x=Weekly$Year,y=Weekly$Volume,main="Year to Volume",xlab="Year",ylab="Volume(billions)")
plot(x=Weekly$Year,y=Weekly$Today,main="Year to Today",xlab="Year",ylab="Today(%return for the week)")

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

# (c)
dir = predict(fit_glm, type="response")
dir
pred = rep("Up", 1089)
pred[dir < 0.5] = "Down"
table(pred, Weekly$Direction)
# Accuracy(y_pred = pred, y_true = Weekly$Direction)
# Recall(y_pred = pred, y_true = Weekly$Direction)
# Precision(y_pred = pred, y_true = Weekly$Direction)
```

```r
# (d)
train <- (Weekly$Year < 2009)
train
Weekly_1990_2008 <- Weekly[train, ]
Weekly_2009_2010 <- Weekly[!train, ]
# Weekly_1990_2008
# Weekly_2009_2010
Direction_1990_2008 <- Weekly$Direction[train]
Direction_2009_2010 <- Weekly$Direction[!train]

# Direction_1990_2008
length(Direction_1990_2008)
lag2_glm <- glm(Direction ~ Lag2, data=Weekly_1990_2008, family=binomial)
summary(lag2_glm)

pred_glm <- predict(lag2_glm, type="response", newdata = Weekly_2009_2010)
plot(pred_glm, pch= ifelse(Weekly_2009_2010$Direction=="Down",8,16), main="predicted direction of 2009 to 2010")
abline(h = 0.5, lwd= 1)

pred2 = rep("Up", 104)
pred2[pred_glm < 0.5] = "Down"
pred2
table(pred2,Direction_2009_2010)

# (e)
lag2_lda <- lda(Direction ~ Lag2, data=Weekly_1990_2008, family=binomial)
lda.pred = predict(lag2_lda,Weekly_2009_2010)
table(lda.pred$class,Weekly_2009_2010$Direction)

# (f)
lag2_qda <- qda(Direction ~ Lag2, data=Weekly_1990_2008, family=binomial)
qda.pred = predict(lag2_qda,Weekly_2009_2010)
table(qda.pred$class,Weekly_2009_2010$Direction)

# (g)
train_X <- cbind(Weekly_1990_2008$Lag2)
test_X <- cbind(Weekly_2009_2010$Lag2)
train_Y <- cbind(Weekly_1990_2008$Direction)
set.seed(1)
# predict when k=1
pred_knn_1 <- knn(train_X, test_X, train_Y, k=1)
```

```
table(pred_knn_1, Weekly_2009_2010$Direction)

## Naive Bayes
fit_nb <- naiveBayes(Direction ~ Lag2, data=Weekly, subset=train)
# predict Direction, result : class
pred_nb <- predict(fit_nb, Weekly_2009_2010)
table(pred_nb)
table(pred_nb, Weekly_2009_2010$Direction)
mean(pred_nb == Weekly_2009_2010$Direction)

# (i)
## Logistic Regression
lag2_glm <- glm(Direction ~ Lag2+Lag3+Lag4+Lag5, data=Weekly_1990_2008, family=binomial)
summary(lag2_glm)
pred_glm <- predict(lag2_glm, type="response", newdata = Weekly_2009_2010)
plot(pred_glm, pch= ifelse(Weekly_2009_2010$Direction=="Down",8,16), main="predicted direction of
2009 to 2010")
abline(h = 0.5, lwd= 1)
pred2 = rep("Up", 104)
pred2[pred_glm < 0.5] = "Down"
pred2
table(pred2,Direction_2009_2010)
## LDA
lag2_lda <- lda(Direction ~ Lag2+Lag3, data=Weekly_1990_2008, family=binomial)
lag2_lda
lda.pred = predict(lag2_lda,Weekly_2009_2010)
table(lda.pred$class,Weekly_2009_2010$Direction)
## QDA
lag2_qda <- qda(Direction ~ Lag2+Lag3, data=Weekly_1990_2008, family=binomial)
lag2_qda
qda.pred = predict(lag2_qda,Weekly_2009_2010)
table(qda.pred$class,Weekly_2009_2010$Direction)
## KNN K=2
train_X <- cbind(Weekly_1990_2008$Lag2)
test_X <- cbind(Weekly_2009_2010$Lag2)
train_Y <- cbind(Weekly_1990_2008$Direction)
set.seed(1)
pred_knn_1 <- knn(train_X, test_X, train_Y, k=4)
table(pred_knn_1, Weekly_2009_2010$Direction)
## NaiveBayes
fit_nb <- naiveBayes(Direction ~ Lag3+Lag4, data=Weekly, subset=train)
fit_nb
```

```
# predict Direction, result : class
pred_nb <- predict(fit_nb, Weekly_2009_2010)
table(pred_nb)
table(pred_nb, Weekly_2009_2010$Direction)
mean(pred_nb == Weekly_2009_2010$Direction)

# 11.
?Auto
summary(Auto)
dim(Auto)
str(Auto)
# (a)
mpg_median = median(Auto$mpg)
mpg_median
Auto$mpg01 = rep(0,392)
Auto$mpg01[Auto$mpg > mpg_median] = 1
head(Auto)
summary(Auto$mpg01)

# (b)
cor(Auto$mpg01,na.omit(Auto[-9]))
#pairs(Auto)
plot(Auto$mpg01, Auto$cylinders,main="mpg01 v.s cylinders",pch=16,xlab = "mpg01",ylab = "cylinders")
plot(Auto$mpg01, Auto$displacement,main="mpg01 v.s displacement",pch=16,xlab = "mpg01",ylab =
"displacement")
plot(Auto$mpg01, Auto$horsepower,main="mpg01 v.s horsepower",pch=16,xlab = "mpg01",ylab =
"horsepower")
plot(Auto$mpg01, Auto$weight,main="mpg01 v.s weight",pch=16,xlab = "mpg01",ylab = "weight")
plot(Auto$mpg01, Auto$acceleration,main="mpg01 v.s acceleration",pch=16,xlab = "mpg01",ylab =
"acceleration")
plot(Auto$mpg01, Auto$year,main="mpg01 v.s year",pch=16,xlab = "mpg01",ylab = "year")
plot(Auto$mpg01, Auto$origin,main="mpg01 v.s origin",pch=16,xlab = "mpg01",ylab = "origin")

# (c)
id <- sample(1:dim(Auto)[1], size=dim(Auto)[1]*0.75)
training <- Auto[id,]
testing <- Auto[-id,]

# (d) lda
auto_lda <- lda(mpg01 ~ cylinders, data=training, family=binomial)
auto_lda
lda.pred = predict(auto_lda,testing)
```

```
table(lda.pred$class, testing$mpg01)

# (e) qda
# cor(na.omit(Auto[-9]),na.omit(Auto[-9]))
auto_qda <- qda(mpg01 ~ cylinders+year+acceleration+origin, data=training, family=binomial)
auto_qda
qda.pred = predict(auto_qda,testing)
table(qda.pred$class, testing$mpg01)

# (f) logistic regression
auto_glm <- glm(mpg01 ~ cylinders, data=training, family=binomial)
summary(auto_glm)
pred_glm <- predict(auto_glm, type="response", newdata = testing)
plot(pred_glm, pch= ifelse(testing$mpg01==0,8,16), main="predicted mpg01")
abline(h = 0.5, lwd= 1)
dim(testing)
pred_mpg01 = rep(1, 98)
pred_mpg01[pred_glm < 0.5] = 0
pred_mpg01
table(pred_mpg01,testing$mpg01)

# (g) KNN
train_X <- cbind(training$cylinders)
test_X <- cbind(testing$cylinders)
train_Y <- cbind(training$mpg01)
set.seed(1)
# predict when k=1
auto_pred_knn_1 <- knn(train_X, test_X, train_Y, k=1)
table(auto_pred_knn_1, testing$mpg01)
```