資料科學期末筆記v.3

黃咨瑀

2020/6/12

資料科學簡介

比較

Data Science	Data Analytics	Big Data
從資料中提取出有價值的問題	從資料中萃取資訊分析(統計學)	來自各種來源的大量非結構化或結構化資料。非結構性資料如影片、動畫、音樂;結構性資如EXCEL
面對一堆資料或沒有和問題對應的資料或是沒有紀錄	場景簡單、已知問題	
預測性what WILL happen?	敘述性what HAPPEND?	
Define Hypothesis to test-> gather data->build data model->explore the data-> build and refine analytic models-> **Ascertain goodness of fit**	**Build the Data Model**->define the report->generate SQL commands-> **create report**	
Python、R、SQL	SAS、SPSS、R、Tableau、Apache Spark	Hadoop、NoSQL、Hive

比較

資料工程師	資料科學家	利害關係人
技術-資料倉儲、大數據	演算、說故事、視覺化、統計、數學、機器學習	利害關係人:財務分析等
data operationalize	data optimization	data monetization

資料探勘可分為監督式學習與非監督式學習

Supervised learning 監督式學習

僅有一個y，訓練資料中有正確答案，由輸入物件和預期輸出所組成，而演算法可以由訓練資料中學到或建立一個模式，並依此模式推測新的實例。用Regression 迴歸(真實的'值') 分析
Classification 分類：分兩類（P/N, Yes/No, M/F, Sick/Not sick）/分多類 (A/B/C/D)
Linear Regression 線性迴歸
Logistic Regression 羅吉斯/邏輯迴歸
Decision Trees 決策樹

Unsupervised learning 非監督式學習

數個y，不用提供正確答案，也就是不需要人力來輸入標籤，單純利用訓練資料的特性，將資料分群分組 PCA主成分分析

在R語言中，資料型態有數值 (numeric)、字串 (character)、布林變數 (logic)等

1. 數值:整數（沒有小數點）與浮點數（有小數點）的數值
2. 用雙引號"框起的文字會被儲存為字串格式，若在數字前後加上雙引號，數字也會被儲存為文字形式，無法進行數值的加減乘除等運算
3. 布林變數:用於邏輯判斷，可使用大寫TRUE或T代表真，大寫FALSE或F代表假

錯誤訊息顯示種類

- Message：有可能的錯誤通知，程式會繼續執行
- Warning：有錯誤，但是不會影響太多，程式會繼續執行
- Error：有錯，而且無法繼續執行程式
- Condition：可能會發生的情況

判斷模型是否正確的方法:

1. 視覺化方式(圖表)表示資料採礦模型的精確度

- lift charts累積增益圖:比較每一模型之預測的精確度
- profit chart收益圖:理論上與使用每個模型相關聯的收益增加
- scatter plot散佈圖:比較實際和預測值，用於迴歸模型等

2. classifiaction matrix分類舉證/confusion matrix混淆矩陣:將正確和不正確的預測表格化
3. cross-validation交叉驗證(訓練、測試樣本):以統計方式驗證採礦模型的可靠性

導入->清理->視覺化->轉換->建模型->溝通(markdown)
EDA->Prediction->Visualization

Markdown

Markdown 常用指令

- `echo=FALSE` :輸出時隱藏程式碼，但結果會顯示
- `message = "FALSE"` :程式碼執行的附帶訊息不會含在輸出文件中
- `warning = "TRUE"` :程式碼執行若有錯誤訊息要含在文件輸出
- `eval = "FALSE"` :輸出時程式碼顯示但不執行
- `results = "hide"` :程式碼執行且顯示，但結果不顯示 除了TRUE/FALSE，也可指定哪幾個expression要輸出，expression不是程式行，而是一個完整的程式表達。如：eval=c(1,5)是指第1和第5個程式表達要輸出

EDA(Exploratory Data Analysis)探索性資料分析

定義

發現問題(被解釋變數Y)，認識資料特性
探索式資料分析包括**分析各變數間的關聯性，看是否有預料之外的有趣發現，或是觀察資料內容是否符合預期，若否，檢查資料是否有誤，最後檢查資料是否符合分析前的假設**。由上述可知，探索式資料分析通常不需要嚴謹的假設和細節呈現，主要功能還是『觀察』資料的特性。

Case 1.小費

使用 = 設定變數，此時變數名稱必須在左側

```
tips=read.csv("./data/tips.csv") #“.”是目前所在路徑。令tips為名稱
head(tips) #看前六筆資料
tail(tips) #後六筆資料
```

Summary statistics

```
summary(tips) #對全部資料做摘要
summary.factor(tips[,"smoker"]) #對分量做資料摘要
tips[,"smoker"] #Run此factor後可以在最下面那行發現它是以level的方式儲存
y=tips[,1] #定義第一欄的資料(即total bills，為一連續資料(有小數點))合併給一個物件y
mean(y)
var(y)
sd(y)
moments::skewness(y)#偏態
moments::kurtosis(y)#峰態
t.test(y)#常態性檢定、常態分佈檢定
```

Pivot table

```
table(tips$day,tips$size)# 列先行後。$:Tips中的day、tips中的size
table(tips$day,tips$size,tips$sex)# 兩個表格呈現
# 原始資料~類別變數
aggregate(tip~time,    data=tips,mean) # 用time分群tip變數，分類完之後取平均數
aggregate(tip~time+day,data=tips,mean) # 用day把資料再分群一次
```

cbind:合併列(兩個列數要相同);rbind:合併行(兩個行數要相同)

```
aggregate(cbind(tip,total_bill)~day,     data=tips,mean) # grouped by day
aggregate(cbind(tip,total_bill)~day+time,data=tips,mean) #re-grouped by time
```

Case 2.航班

```
dat=read.csv("./data/flights.csv")
unique(dat)# unique: unique returns a vector, data frame or array like x but with duplicate elem
ents/rows removed 判斷對象的每個取值是否重複，如unique(c(1,1,2,3))返回1 2 3
tail(dat)
```

Carrier航空公司名稱;origin起飛機場

Subset

EDA:提問(1):How many trips have been made in 2014 from JFK airport? 先抓出從JFK出發的資料，看有多少列

```
ans1=subset(dat, origin == "JFK")
nrow(ans1)
```

```
    [1] 81483
```

故在這段期間內為81483班次

提問(2):How many trips have been made in 2014 from JFK airport in the month of June?承上題，指定為6月份?
一樣抓出來，L代表宣告為整數

```
ans2=subset(dat, origin == "JFK" & month == 6L)
nrow(ans2)
```

```
    [1] 8422
```

故在這段期間內為81483班次
提問(3):誤點班次數?

```
dat$Delay=sum(dat$arr_delay + dat$dep_delay)
ans3=subset(dat, Delay < 0)
nrow(ans3)
```

```
    [1] 0
```

線性迴歸做預測

A. The conventional statistical analysis

用在因變數Y連續時 ### 1. Regression– Estimation of conditional mean

Y=sales(預測變數銷售量), x=advertisement(廣告支出)

向量轉成矩陣:as.matrix
文字轉成factor: as.factor

summary(output1)的這個物件，他有哪些名稱
把想要的名稱叫出來，就用$加上名稱的前幾個字即可

在R中，最基本的簡單線性迴歸分析為 `lm()` ，使用方法為 `lm(formula,data=資料名稱)` ，搭配formula使用，formula的撰寫方法為：y~x1＋x2＋….

```
Eq=as.formula("Sales~Adv")#將文字"y~x"轉換成(用as.formula)數學公式，而非字串
output1=lm(Eq,data=dataUsed)#線性回歸linear model
summary(output1)$coef
```

```
                 Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 134.13993781 7.536574679 17.798528 5.967817e-43
Adv           0.09612449 0.009632366  9.979322 2.941980e-19
```

Y的變異只有**0.3346**能夠被**X**的變異解釋，有剩下**60%**無法解釋
平均上，廣告支出高於平均支出**0.09612449**單位時，**Sales**會高於平均**Sales 10%**
但只有**0.3346**的機率會成功，可性度不高

```
summary(dataUsed)
```

```
     Sales            Adv              airplay         attract     
 Min.   : 10.0   Min.   :   9.104   Min.   : 0.00   Min.   : 1.00  
 1st Qu.:137.5   1st Qu.: 215.918   1st Qu.:19.75   1st Qu.: 6.00  
 Median :200.0   Median : 531.916   Median :28.00   Median : 7.00  
 Mean   :193.2   Mean   : 614.412   Mean   :27.50   Mean   : 6.77  
 3rd Qu.:250.0   3rd Qu.: 911.226   3rd Qu.:36.00   3rd Qu.: 8.00  
 Max.   :360.0   Max.   :2271.860   Max.   :63.00   Max.   :10.00  
```

廣告支出花費**614.412**元，預測銷售量為**193.2**單位
若想要有較高的銷售量，就要高於平均**614.412**的廣告支出

Interpretation of estimates

```
dev.new() # Device new: 在畫圖的時候可以另外開一個視窗，而不是直接顯示在下方
par(mfrow=c(2,2))# 圖形切成2列乘2行，共四格
plot(output1,which=c(1:3,5))# 取第1,2,3,5的output項
par(mfrow=c(1,1))#回到1乘1，免得後面的圖都會被切成2乘2
```

Interpretation of the 4 plots

畫出期望值、信賴區間、原始資料
fit:預測值，lwr:下界，upr上界
Lines是把資料點用線連接，產生上下界線
lty是線的種類，lwd是線的粗細
在左上角增加說明圖例框，bty是圖例框是否畫出(是=o，否=n)

```
Pred=predict(output1, interval="confidence") #將回歸的結果output1拿來做predict，在預測(predict)出來的結果中添加信賴區間
dev.new()# 開出新視窗
with(plot(Sales~Adv),data=dataUsed)# Plot(X~Y)  圖形上會顯示成X是X軸的值和名稱，Y是Y軸的值和名稱
abline(output1,col="blue")# Add Straight Lines to a Plot:畫上線
with(lines(Pred[,"lwr"] ~ Adv, lwd=0.05, lty=4, col=2),data=dataUsed)
with(lines(Pred[,"upr"] ~ Adv, lwd=0.05, lty=4, col=2),data=dataUsed)
legend("topleft", c("regression line", "low", "upper"), lty=c(1,4,4), lwd=0.05, bty="n")
```

abline 函數為控制繪圖的線，可以將資料進行迴歸後輸入

- cex文字大小
- lty畫線類型
- abline:函數為控制繪圖的線
- col:為控制顏色
- h:為根據 y 軸的水平線
- v:為根據 x 軸的垂直線
- lwd:寬度
- a 與 b 之間是相互關連， a 為 y 軸起始，b 則為斜率

How to evaluate the performance of prediction?

Error:殘差(真實-預測)，即RMSE(root mean square error=開根號的E乘以e^2)
Sqrt:開根號
三個常用的預測診斷，越小越好，且不可互相比，ex. 要跟不同模型的rmse比
/100是百分比
MAE(Mean Absolute Error)(預測錯誤的期望值)
MAPE(Mean absolute percentage error)(預測錯誤站原始資料的百分比)

```
newData1=data.frame(Y=dataUsed$Sales, EY=predict(output1),error=output1$residuals)#將三筆資料整一newdata1
Error=newData1$error
sqrt(mean(Error^2))
Performance1=forecast::accuracy(newData1$Y,newData1$EY)# 用套件forecast中的函數accuracy，給兩筆資料:真資料與預測資料計算出五個指標
Performance1[,c("RMSE","MAE","MAPE")]/100
```

2. ANOVA

看output1還是output2哪個模型比較好:虛無假設是1比較好，對立假設是2
如果一個統計檢驗的結果拒絕虛無假設（結論不支持虛無假設），而實際上真實的情況屬於虛無假設，那麼稱這個檢驗犯了第一類錯誤。反之，如果檢驗結果支持虛無假設，而實際上真實的情況屬於備擇假設，那麼稱這個檢驗犯了第二類錯誤。通常的做法是，在保持第一類錯誤出現的機會在某個特定水平上的時候（即顯著性差異值或α值），儘量減少第二類錯誤出現的機率

```
output2=lm(Sales~.,data=dataUsed)# ~後的.代表sales中的變數(除了sales之外的其他所有變數)我都要有，與前面output1只有廣告支出不同，2是把所有變數都放入
anova(output1,output2)
```

```
    Analysis of Variance Table

    Model 1: Sales ~ Adv
    Model 2: Sales ~ Adv + airplay + attract
      Res.Df    RSS Df Sum of Sq      F    Pr(>F)
    1    198 862264
    2    196 434575  2    427690 96.447 < 2.2e-16 ***
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

F檢定小於0.05 拒絕虛無假設

3. Save output coefficient table

```
coef_output1=summary(output1)$coef
write.csv(coef_output1,file="./output/data/coef_output1.csv")# 存output至./output/data/coef_output1.csv中
papeR::prettify(summary(output1), confint = FALSE)# 美化後儲存
coef_output2=papeR::prettify(summary(output1), confint = FALSE)
write.csv(coef_output2,file="./output/data/coef_output2.csv")
```

papeR的套件中有prettify函數，美化後不會出現p那種醜數字，美化前後如下:

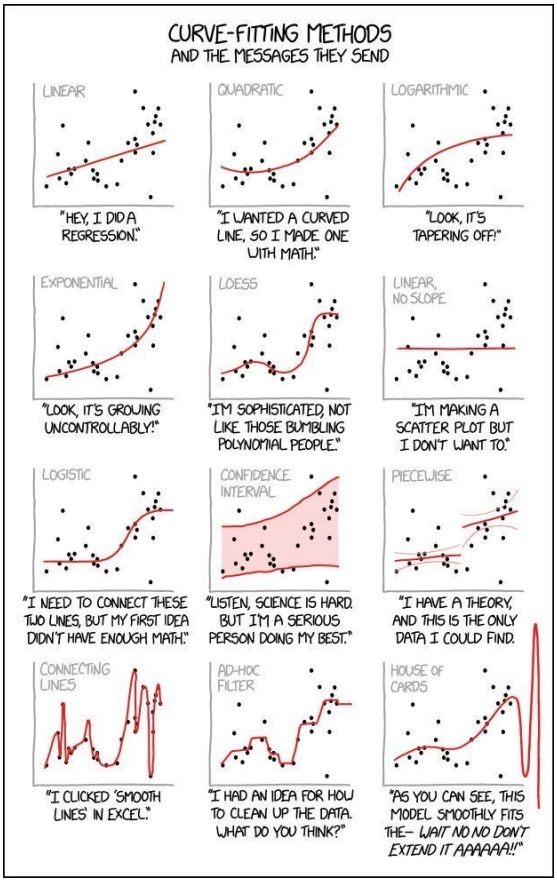
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	134.1399	7.536575	17.79853	5.97E-43
Adv	0.096124	0.009632	9.979322	2.94E-19

unprettifyplot

		Estimate	Std. Error	t value	Pr(>|t|)
1	(Intercept)	134.1399	7.536575	17.79853	<0.001
2	Adv	0.096124	0.009632	9.979322	<0.001

prettifyplot

使用不同的函數會有以下不同的圖形



Compare the forecast performance of output2 with output1.

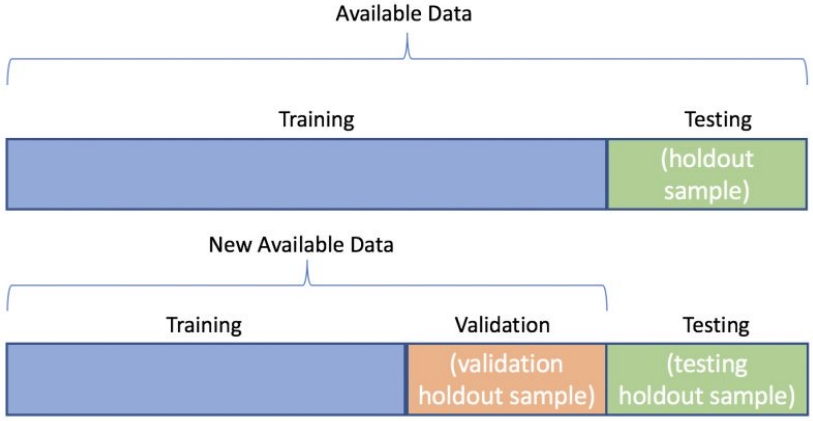
```
newData2=data.frame(Y=dataUsed$Sales, EY=predict(output2))# 和output1的performance2相比
Performance2=forecast::accuracy(newData2$Y,newData2$EY)
Performance2[,c("RMSE","MAE","MAPE")]/100
```

```
       RMSE       MAE      MAPE
  0.4661408 0.3665269 0.2211853
```

預測精確度是用預測錯誤的指標來看，RMSE, MAE, MAPE越小越好
故因Output1的錯誤指標皆>2，所以2較好

B. The data science searches the best model

用**data**來訓練模型~資料科學=/=統計學
70%:隨機取樣訓練樣本training sample(y1=a+bx1)
15%:確認樣本validation sample(y2=a+bx2)
15%:測試樣本testing sample (y3=a+bx3)



用最多的資料訓練模型，將模型訓練完的結果預測資料以外的部分(非估計以外，即validation sample)
重複做CV(cross validation)交叉驗證(隨機取樣和validation sample)，降低overfitting
準確度若training > validation，則為Overfitting;若training < validation，則為underfitting
理想為兩者差不多，或是像validation稍微大training一點點，若超過一倍則要重新建模

```
dataUsed$subSample = trainingSamples(dataUsed, Training=0.7, Validation=0.15)# 等號左邊是新增欄位，右邊是新增欄位的內容。要在右邊新增一個現有資料表的新欄位subsample就在前面加上$，此欄位就會由traingsamples函數執行，training70%，validation和testing15%(比例可自行調配)
table(dataUsed$subSample)
```

```
      Testing   Training Validation
           30        140         30
```

```
trainingSample=subset(dataUsed, subSample=="Training")# 將dataUsed取子樣本(subset)，只要最後一欄subsample=training就把它挑出來，並命名為trainingsample
validationSample=subset(dataUsed, subSample=="Validation")
testSample=subset(dataUsed, subSample=="Testing")# 同trainingsample，subsample三個字串所組成(training, validation, testing)
```

資料科學3 STEPS

Step 1. Training your regression

```
output_training=lm(Eq,data=trainingSample)# Eq是銷售額sales對廣告支出adv的迴歸，data用traingsample而非原始full資料
summary(output_training)# 迴歸估出來的結果是output_training
```

Step 2. Prediction of validation sample

用validationSample產生預測，看預測的表現如何

```
Pred_validation=predict(output_training,validationSample)
newDataValid=data.frame(Y=validationSample$Sales, EY=Pred_validation)
forecast::accuracy(newDataValid$Y,newDataValid$EY)[,c("RMSE","MAE","MAPE")]/100
```

```
       RMSE       MAE      MAPE
  0.6802736 0.5709892 0.3227935
```

用validation sample產生預測，output_training就是估出來的結果(Sales=134.45158+0.9686Adv)
不錯的話就會用traingsample估計的結果 (即output_training)來testsample
沒有通過validation檢測的話，是不做step3的

Step 3. Generate prediction for decision making

所有模型都跑完後，挑最好的來testSample
Validation sample不錯的話就會用traingsample估計的結果 (即output_training)來test sample

```
Pred_testing=predict(output_training,testSample)
newDataTest=data.frame(Y=testSample$Sales, EY=Pred_testing)
forecast::accuracy(newDataTest$Y,newDataTest$EY)[,c("RMSE","MAE","MAPE")]/100
```

```
      RMSE       MAE      MAPE 
 0.5646992 0.4223502 0.2376709 
```

比validation sample好
跑完step3也沒好到哪的話，就要修正整個模型

How to augment your training models, and select the best one?

上述的三步驟只跑一條回歸，但AlbumSales中有三個解釋變數(adv,airplay,attract)，到底哪一個好?應該把所有可能性展開後全部跑一遍，再挑出最好的→The data science searches the best model(以此來預測ex.消費者行為、股票價格報酬率等等)

1.定義解釋變數

```
indeps=names(dataUsed)[-c(1,5)]
indepEQ=paste(indeps,collapse = "+")
indep1=paste(indeps[-1],collapse = "+")
indep2=paste(indeps[-2],collapse = "+")
indep3=paste(indeps[-3],collapse = "+")
```

當中的

```
names(dataUsed)
```

```
[1] "Sales"     "Adv"       "airplay"   "attract"   "subSample"
```

但我們要解釋變數，故不要sales和subsample，寫成

```
names(dataUsed)[-c(1,5)]
```

```
[1] "Adv"     "airplay" "attract"
```

```
paste(indeps,collapse = "+")# 並定義為物件indeps,Paste:把物件都黏起來
```

```
[1] "Adv+airplay+attract"
```

2.建立方程式

一個變數的迴歸—一個Y對一個X

```
Eqs=paste0("Sales ~ ",indeps)
Eqs0=paste0("Sales ~ ",indepEQ)
Eqs1=paste0("Sales ~ ",indep1)
Eqs2=paste0("Sales ~ ",indep2)
Eqs3=paste0("Sales ~ ",indep3)
```

```
paste0("Sales ~ ",indeps)
```

```
    [1] "Sales ~ Adv"     "Sales ~ airplay" "Sales ~ attract"
```

共三個方程式

```
paste0("Sales ~ ",indepEQ)
```

```
    [1] "Sales ~ Adv+airplay+attract"
```

三個變數都使用，產生對sales的預測
以下為任兩個變數對sales的預測

```
Eqs1=paste0("Sales ~ ",indep1)
Eqs2=paste0("Sales ~ ",indep2)
Eqs3=paste0("Sales ~ ",indep3)
```

還有其他情況(像是最後一行):

```
Eqs01=paste0("Sales ~ ",paste0("(",indepEQ,")^3"))
Eqs02=paste0("Sales ~ ",paste0("(",indepEQ,")^2"))
Eqs11=paste0("Sales ~ ",paste0("(",indep1,")^2"))
Eqs21=paste0("Sales ~ ",paste0("(",indep2,")^2"))
Eqs31=paste0("Sales ~ ",paste0("(",indep3,")^2"))
paste0("Sales ~ ",paste0("(",indepEQ,")^3"))
```

```
    [1] "Sales ~ (Adv+airplay+attract)^3"
```

以上為多交互項

$$y \sim (x1+x2+x3)^3$$
$$y = (x1+x2+x3)+(x1\times x2+x1\times x3+x2\times x3)+(x1\times x2\times x3)$$

$$y \sim (x1+x2+x3)^2$$
$$y = (x1+x2+x3)+(x1\times x2+x1\times x3+x2\times x3)$$

以上是一個展開，若改成2次則為

$$y \sim (x1+x2+x3)^3$$
$$y = (x1+x2+x3)+(x1\times x2+x1\times x3+x2\times x3)+(x1\times x2\times x3)$$

$$y \sim (x1+x2+x3)^2$$
$$y = (x1+x2+x3)+(x1\times x2+x1\times x3+x2\times x3)$$

把所有方程式堆起來

```
EQ=c(Eqs,Eqs0,Eqs1,Eqs2,Eqs3,Eqs01,Eqs02,Eqs11,Eqs21,Eqs31)
EQ
```

```
     [1] "Sales ~ Adv"                    "Sales ~ airplay"
     [3] "Sales ~ attract"                "Sales ~ Adv+airplay+attract"
     [5] "Sales ~ airplay+attract"        "Sales ~ Adv+attract"
     [7] "Sales ~ Adv+airplay"            "Sales ~ (Adv+airplay+attract)^3"
     [9] "Sales ~ (Adv+airplay+attract)^2" "Sales ~ (airplay+attract)^2"
    [11] "Sales ~ (Adv+attract)^2"        "Sales ~ (Adv+airplay)^2"
```

共12條方程式

3.執行迴歸產生預測

資料科學STEP1 & 2
取出第一條方程式

```
i=1
output_training=lm(as.formula(EQ[i]),data=trainingSample)
Pred_validation=predict(output_training,validationSample)
newDataValid=data.frame(Y=validationSample$Sales, EY=Pred_validation)
forecast::accuracy(newDataValid$Y,newDataValid$EY)[,c("RMSE","MAE","MAPE")]/100
EQ[1]
Eqs01# 若想要看 Eqs01 的結果，執行Eqs01得到Sales ~ (Adv+airplay+attract)^3
EQ# 執行EQ找到Sales ~ (Adv+airplay+attract)^3是在第8個
i=8# 將i改成=8
output_training=lm(as.formula(EQ[i]),data=trainingSample)
Pred_validation=predict(output_training,validationSample)
newDataValid=data.frame(Y=validationSample$Sales, EY=Pred_validation)
forecast::accuracy(newDataValid$Y,newDataValid$EY)[,c("RMSE","MAE","MAPE")]/100
EQ[i]# 確認一下，執行EQ[i]為Sales ~ (Adv+airplay+attract)^3
```

```
summary(output_training)# 最後，可以用summary來看一下output_training
```

```
Call:
lm(formula = as.formula(EQ[i]), data = trainingSample)

Residuals:
     Min       1Q   Median       3Q      Max
-151.903  -31.018    2.581   31.858  127.447

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.389e+02  1.008e+02   1.378   0.1705
Adv                 -2.667e-03  1.789e-01  -0.015   0.9881
airplay             -3.522e+00  3.358e+00  -1.049   0.2963
attract             -1.334e+01  1.424e+01  -0.937   0.3505
Adv:airplay          3.975e-03  5.532e-03   0.718   0.4738
Adv:attract          1.291e-02  2.515e-02   0.513   0.6085
airplay:attract      1.004e+00  4.682e-01   2.144   0.0338 *
Adv:airplay:attract -5.738e-04  7.821e-04  -0.734   0.4645
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49.36 on 132 degrees of freedom
Multiple R-squared:  0.6502,    Adjusted R-squared:  0.6317
F-statistic: 35.05 on 7 and 132 DF,  p-value: < 2.2e-16
```

可以看到intercept的前三個獨立，第4、5個兩兩相乘，最後三個相乘
程式語言的相乘用冒號：做運算 如此反覆操作**12**個模型，看此**12**個模型那個預測率最好

自動化迴圈

上面方法太慢，故要將i=1-12做自動化迴圈 定義ACC=空集合
i=1到EQ的長度
重複步驟一和二
不過這次我們希望將預測的結果蒐集起來，故令ACC0
蒐集方式是使用ACC=rbind *Rbind即為rowbind，用列把它疊起來*

>有時需要在資料框新增一列，或新增一行，可以利用資料組合函數完成
1.Row 列的組合 rbind()
2.Column 行的組合 cbind()

單純只看ACC(> ACC)的話如下:

```
          RMSE       MAE       MAPE
ACC0 0.6802736 0.5709892 0.3227935
ACC0 0.6209194 0.5180576 0.2672982
ACC0 0.7501500 0.5854551 0.3097612
ACC0 0.4662889 0.3577408 0.2211348
ACC0 0.6089666 0.4790638 0.2425249
ACC0 0.6135834 0.4938293 0.2820967
```

單純只看ACC(> ACC)的話，跑出來會有一堆ACC0，但我們其實想知道那是第幾次，因此去除ACC那行，置換成1到length(EQ)的數字
此時ACC0會被置換成編碼

```
rownames(ACC)=1:length(EQ)
head(ACC)
```

```
        RMSE       MAE       MAPE
1 0.6802736 0.5709892 0.3227935
2 0.6209194 0.5180576 0.2672982
3 0.7501500 0.5854551 0.3097612
4 0.4662889 0.3577408 0.2211348
5 0.6089666 0.4790638 0.2425249
6 0.6135834 0.4938293 0.2820967
```

現在要抓出最好的:三個橫向比是沒有意義的，要垂直的比

挑選模型

法1.Select model by accuracy stnadard-根據單一指標挑最好的模型

```
which.min(ACC[,"RMSE"])#12個模型當中RMSE最小的是在第12個
```

```
12
12
```

```
m1=which.min(ACC[,"RMSE"])#令他為m1
m2=which.min(ACC[,"MAE"])#MAE最小的是在第4個，令他為m2
m3=which.min(ACC[,"MAPE"])#MAPE最小的是在第7個，令他為m3
EQ[m3]#12個模型當中MAPE最小的是[1] "Sales ~ Adv+airplay"
```

```
[1] "Sales ~ Adv+airplay"
```

法2. Select model by average-三個指標相加取平均數最小的模型

Apply是計算資料表的一個函數，可計算列行兩種維度

```
        1         2         3         4         5         6         7         8
0.5246854 0.4687584 0.5484554 0.3483882 0.4435184 0.4631698 0.3438918 0.3720168
        9        10        11        12
0.3718058 0.4613459 0.4576332 0.3433790
```

```
12
12
```

如果今天採用的模型不是迴歸(lm)ex.決策數、隨機森林，只需要給資料就好，不需要給equation。但因為今天是迴歸(統計)，一定要 `as.formula` ，否則不能做計算

IMDb抓資料

透過寫好的source code以及目的網站的連結

```
get_movie_rating(breakingbad_url)#電影評分
get_movie_genre(breakingbad_url)#電影種類
get_movie_cast(breakingbad_url)#卡司
myurl=get_movie_poster(breakingbad_url)
library(magick)
POSTER=image_read(myurl)
print(POSTER)#讀取海報圖片
```

探索式資料分析-資料清理

String是字符串，可用於記錄瑣碎訊息（比如發現UFO者的口頭描述內容）。Factor是用於給一行記錄做“分類標記”，比如人的性別factors可以設置為“男”、“女”。對於Factor類型屬性，R語言可以自動統計數據的factor水平（level），比如，男，有多少，Mon有多少等

`stringsAsFactors=F` 意味著，“在讀入數據時，遇到字符串之後，不將其轉換為factors，仍然保留為字符串格式

用 `read.table` 和比如 `read.csv` 、 `read.delim` ，R會自動把字符串string的列辨認成factor。比如你有一個全制班成績數據集,第一列名字,第二列性別,第三列語文成績,第四列數學成績。那麼第一列和第二列如果不告訴R，stringsAsFactors=FALSE，那麼R就把這兩列認成因子模式factor了

BASIC PIVOTTABLE SUMMARIES

在R裡面，判斷某個值(或向量)，是否存在於另一個向量之中，會使用 %in% 的符號

- `x %in% c(1,2,3,4,5)` 值是否存在向量內
- `y %in% c(1,2,3,4,5)` 向量內的各值，是否存在於另一個向量內

dplyr使用以下函數分析整理資料：

- `select()`：選要分析的欄位，欄位子集 (Column)
- `filter()`：選要分析的觀察值，觀察值子集 (Row)
- `mutate()`：增加新欄位
- `summarise()`：計算統計值
- `group_by()`：分組依據，依照類別變數分
- `arrange()`：觀察值排序
- `rename()`：欄位重新命名
- `%>%`

filter to keep three states.

使用 `filter()` 可選要分析的觀察值，也就是針對列做子集，使用方法為 `filter(資料名稱,篩選條件)`

```
library(dplyr)  #資料處理分析套件
basic_summ0 = filter(mprices, state %in% c("California", "New York", "Illinois")) #取出部分資料，子資料 subset of data;在mprice資料內，要state中的加州、紐約州、伊利諾州
```

set up data frame for by-group processing.

在quality 和state 做group的戳記
使用 dplyr 套件中基礎函數之後的輸出是一種叫做 tibble 的改良式資料框(32508列，10行)

```
basic_summ1 = group_by(basic_summ0, quality, state) #對特定欄位做戳記(mark)
```

calculate the three summary metrics

對已戳記資料做摘要:加總amount(根據**quality,**、**state**交叉表加總);根據quality、state平均ppo…

```
basic_summ = summarise(basic_summ1,
                       sum_amount = sum(amount),
                       avg_ppo = mean(ppo),
                       avg_ppo2 = sum(price) / sum(amount))
```

戳記差別

Replacing basic_summ1 by basic_summ0 as basic_summ00
用 `head(basic_summ0)` 和
`head(as.data.frame(basic_summ1))` 可看出資料結構都一樣
有無group_by在顯示上是沒有任何別的，僅是做了戳記，但此戳記僅電腦看的到
下面是無戳記(basic_summ)和有戳記的(basic_summ00)

```
# A tibble: 9 x 5
# Groups:   quality [3]
  quality        state      sum_amount avg_ppo avg_ppo2
  <chr>          <chr>           <dbl>   <dbl>    <dbl>
1 high quality   California      5495.    278.     227.
2 high quality   Illinois        1538.    376.     311.
3 high quality   New York        2252.    375.     306.
4 low quality    California       356.    275.     190.
5 low quality    Illinois         119.    227.     143.
6 low quality    New York         171.    350.     176.
7 medium quality California      6054.    212.     166.
8 medium quality Illinois        1351.    306.     218.
9 medium quality New York        2316.    288.     223.
```

```
  sum_amount  avg_ppo avg_ppo2
1   19650.85 281.9319 221.0306
```

group_by不會改變資料的樣貌，只是會在資料內做分組的動作

TRANSPOSED SUMMARIES轉置摘要

用 `library(reshape2)` 換形狀(轉置)的套件
decast: 用state(列) ~ quality(行)交叉，重新打造，以 `avg_ppo2` 為值做摘要表

```
library(reshape2)
basic_summ_t = dcast(basic_summ, state ~ quality, value.var = "avg_ppo2")
```

melt:
1. 先recycle state(加->紐->伊)，再recycle quiality(高->低->中)
2. 垂直疊“avg_ppo2”和“sum_amount”兩個欄位

```
basic_summ_t2 = melt(basic_summ, id.vars = c("state","quality"),
                     measure.vars = c("avg_ppo2", "sum_amount"))
```

ADVANCED BY-GROUP PROCESSING

first set up the data frame for by-group processing, by quality
使用 `mutate()` 增加新欄位，如新增新欄位 `ppo_rank` ，欄位值為依據ppo_rank做排序，則指令如下

```
basic_summ_rank = group_by(basic_summ, quality)
basic_summ_rank = mutate(basic_summ_rank, ppo_rank = rank(avg_ppo2))
```

Generalized Linear Models廣義線性迴歸模型

用在y=(0,1)

簡介

資料處理方式和Y的特性有有關

因變量Y的特性

- binary:{0,1}二元(下面以此為主)
- ordered:{1,2,3,4….}有排序ex.問卷調查不滿意、滿意、非常滿意
- count:{100,99,410,1534…}ex.卜瓦松分布，y不連續

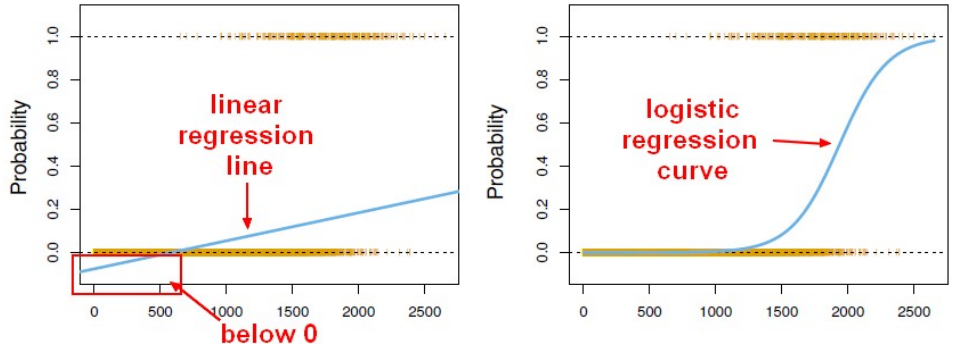
而Logistic Regression常用在y為二元變數（非0即1）(類別/間斷變數)，如：生病/沒生病;錄取/不錄取 所有的估計，都是在估計被觀察變數的期望值

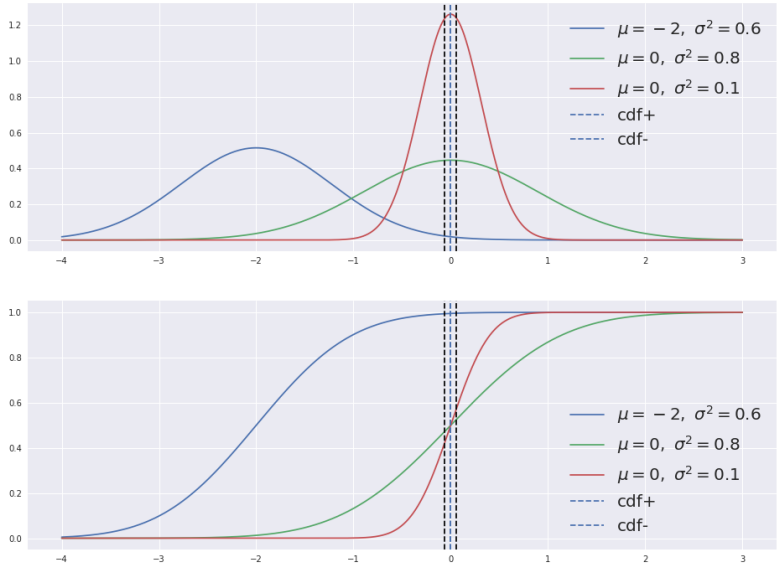
$$y_i = E(y_i) + e_i = \frac{\sum y_i}{n} + e_i$$

樣本期望值

$$y_i = E(y_i \mid x_i) + e_i = \beta_0 + \beta_1 x_i + e_i$$

一般線性回歸有無法預測到的部分，因此要做一般化的轉化成為廣義線性迴歸
線性回歸-會有機率超出0-1的x，廣義線性回歸將機率控制在0-1之間產生非線性累積線性函數CDF





廣義線性迴歸PDF&CDF

Y服從指數分布的類型

`glm()`，使用方法與 `lm()` 類似，包括了線性迴歸模型和邏輯迴歸模型。 如果需要修改預設模型，可設定family參數：

- `family="gaussian"` 常態分布的密度函數
- `family="binomial"` 邏輯迴歸模型(二項分布)
- `family="poisson"` 卜瓦松迴歸模型(次數分佈)

案例: Canadian Charitable Society自動扣款 monthgive

總共有1600筆資料，各欄位意義見Canadian Cancer Society's frequent giving program case explained (https://github.com/ZiYu-Huang/108-2-data-science/blob/master/Canadian%20Cancer%20Society's%20frequent%20giving%20program_case%20explained.pdf)

比較:Statistical Analysis統計分析

```
Eq0=as.formula(MonthGive ~  AveDonAmt + AveIncEA)#先定義formula
output = glm(Eq0,family=binomial(logit), data=dataset)
summary(output)$coef
```

```
                  Estimate   Std. Error   z value     Pr(>|z|)
(Intercept) -5.053748e-01 1.656442e-01 -3.050966 2.281061e-03
AveDonAmt    2.585376e-02 2.589574e-03  9.983788 1.794816e-23
AveIncEA    -3.696074e-06 2.611633e-06 -1.415235 1.569997e-01
```

即MonthGive=-0.5+0.02AveDonAmt-0.0000003AveIncEA+e(0.02是顯著的)
線性迴歸(LM)對0.02的解釋:AveDonAmt增加一單位，MonthGive會增加0.02單位;
而GLM對0.02的解釋:不像LM解釋相對變化，只能用正負號，即AveDonAmt越多，MonthGive=1的機率要大(y=[0,1]no,yes)

Data Science: Predictive Analytics

在原來的dataset中新增變數subsample，subsample中會產生文字串training、validation…再根據這些文字串取出子樣本做分析

y=f(x)->f:函數(轉換x->y)，定義X如何計算。測量方法牽扯到模型和預測

1. Model Building by glm

先取出subset(都是training)，要用訓練樣本而非所有資料(dataset)，因為是抽樣的，所以每次的結果會不太一樣

```
dataTrain=subset(dataset,subSample=="Training")
linearCCS=glm(MonthGive ~ Age20t29 + Age70pls + AveDonAmt + AveIncEA + DonPerYear + EngPrmLang +
FinUnivP+ LastDonAmt+Region + YearsGive, family=binomial(logit), data=dataTrain)
summary(linearCCS)
```

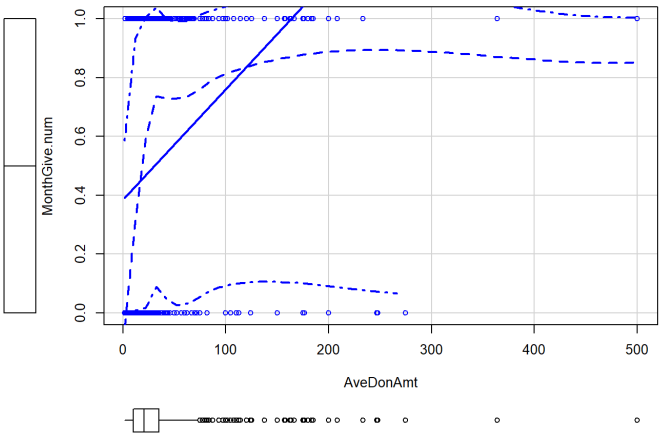
AveDonAmt < 0:平均捐款金額**AveDonAmt**越多，成為月扣款**Mothgive=1**的機會越小，且不顯著
DonPerYear> 0:每年捐的次數**DonPerYear**越多，成為月扣款**Mothgive=1**的機會越大，且非常顯著，可能是因為捐款額度不一定是有錢人，但應該是樂於分享奉獻的人

Why there is no relationship between AveDonAmt (Average Donation Amount) and MonthGive (Become a Monthly Giver)?
可能:非線性被假設成線性

Check non-linearity

方法一

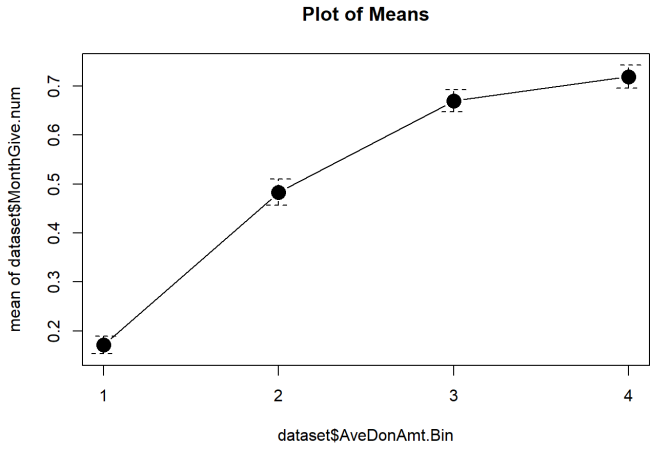
畫散佈圖scatter plot
在原來的子資料中新增一欄變數MonthGive.num，又因為MonthGive是yes/no沒辦法畫， `as.integer` 先將之轉成(1,2)，再-1成(0,1)



regLine是線性那條，smooth是虛線(平滑曲線)，兩條差蠻多的:告訴我們最佳關聯性應該是這樣，AveDonAmt為連續非線性

方法二

畫出每一群中的平均數(plot of means)將橫軸AveDonAmt平均分成四個級距(`bins=4`)



由圖可看出非直線，增加比例非固定，為非線性
平均捐款金額越多，mothgive會增加，但不是固定比例增加

Solution for nonlinearity

取log Use log(AveDonAmt), instead of AveDonAmt
因為此種非線性是刻度問題，要將金額幾百幾千的大數字AveDonAmt縮小到(0,1)可以用Log解決
新增一筆將原始資料取Log的資料
將之前Eq中的AveDonAmt換成log.AveDonAmt

```
Call:
glm(formula = MonthGive ~ Age20t29 + Age70pls + log.AveDonAmt +
    AveIncEA + DonPerYear + EngPrmLang + FinUnivP + LastDonAmt +
    Region + YearsGive, family = binomial(logit), data = dataTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5956  -0.9222   0.2488   0.9572   2.0922

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.459e+00  8.297e-01  -5.374 7.70e-08 ***
Age20t29      -7.051e-01  1.661e+00  -0.424   0.6712
Age70pls       1.332e-01  8.665e-01   0.154   0.8778
log.AveDonAmt  1.239e+00  1.281e-01   9.666  < 2e-16 ***
AveIncEA      -1.021e-05  5.382e-06  -1.896   0.0579 .
DonPerYear     9.101e-01  1.812e-01   5.022 5.12e-07 ***
EngPrmLang     6.676e-01  6.256e-01   1.067   0.2859
FinUnivP       7.635e-01  9.045e-01   0.844   0.3986
LastDonAmt    -4.256e-03  2.462e-03  -1.729   0.0839 .
RegionR2       3.509e-01  2.290e-01   1.532   0.1255
RegionR3       4.518e-01  2.308e-01   1.957   0.0503 .
RegionR4      -3.065e-01  2.943e-01  -1.042   0.2976
RegionR5      -2.552e-01  2.249e-01  -1.134   0.2566
RegionR6      -2.278e-01  3.571e-01  -0.638   0.5235
YearsGive      3.200e-02  2.468e-02   1.297   0.1947
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1552.6  on 1119  degrees of freedom
Residual deviance: 1259.6  on 1105  degrees of freedom
AIC: 1289.6

Number of Fisher Scoring iterations: 4
```

可看出log.AveDonAmt*變成>0，且是顯著的*從0變1的機率是遞增的

What is the impact of Regions?
Monthly Giver vs. 6 Regions

Check importance of Region

```
Analysis of Deviance Table (Type II tests)

Response: MonthGive
           LR Chisq Df Pr(>Chisq)
Age20t29      0.000  1  0.9821847
Age70pls      0.015  1  0.9033073
AveDonAmt     0.060  1  0.8064124
AveIncEA      6.151  1  0.0131327 *
DonPerYear   43.609  1   4.01e-11 ***
EngPrmLang    0.158  1  0.6910351
FinUnivP      2.898  1  0.0887047 .
LastDonAmt   14.853  1  0.0001162 ***
Region       16.442  5  0.0056897 **
YearsGive     6.060  1  0.0138268 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Call:
glm(formula = MonthGive ~ Age20t29 + Age70pls + AveDonAmt + AveIncEA +
    DonPerYear + EngPrmLang + FinUnivP + LastDonAmt + Region +
    YearsGive, family = binomial(logit), data = dataTrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2834  -0.9876   0.0616   1.0733   1.9309

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.341e+00  7.358e-01  -1.823 0.068290 .
Age20t29    -3.583e-02  1.604e+00  -0.022 0.982184
Age70pls     1.007e-01  8.294e-01   0.121 0.903316
AveDonAmt   -1.520e-03  6.185e-03  -0.246 0.805855
AveIncEA    -1.282e-05  5.233e-06  -2.450 0.014305 *
DonPerYear   1.085e+00  1.780e-01   6.093 1.11e-09 ***
EngPrmLang   2.406e-01  6.047e-01   0.398 0.690759
FinUnivP     1.474e+00  8.693e-01   1.695 0.090001 .
LastDonAmt   2.281e-02  6.114e-03   3.731 0.000191 ***
RegionR2     3.703e-01  2.197e-01   1.685 0.091918 .
RegionR3     4.356e-01  2.199e-01   1.981 0.047604 *
RegionR4    -2.593e-01  2.820e-01  -0.920 0.357803
RegionR5    -3.890e-01  2.129e-01  -1.827 0.067669 .
RegionR6    -1.755e-01  3.355e-01  -0.523 0.600924
YearsGive    5.806e-02  2.374e-02   2.446 0.014437 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1552.6  on 1119  degrees of freedom
Residual deviance: 1351.1  on 1105  degrees of freedom
AIC: 1381.1

Number of Fisher Scoring iterations: 5
```

由ANOVA可以知道區域是重要變數，但是下面迴歸glm 卻看不出它統計上的顯著性

Solution for Region

Relabel R2 and R3 as VanFraser, the remaining as others
解法：濃縮區域為兩個，2+3為VanFraser其他為others
承上題的Eq，Region改成Region.New

```
Call:
glm(formula = MonthGive ~ AveIncEA + DonPerYear + LastDonAmt + 
    log.AveDonAmt + Region.New + YearsGive, family = binomial(logit), 
    data = dataTrain)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5725  -0.9419   0.2529   0.9699   2.1183  

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)         -4.096e+00  4.149e-01  -9.871  < 2e-16 ***
AveIncEA            -7.492e-06  3.536e-06  -2.119  0.03409 *  
DonPerYear           9.218e-01  1.797e-01   5.129 2.92e-07 ***
LastDonAmt          -4.004e-03  2.433e-03  -1.646  0.09974 .  
log.AveDonAmt        1.233e+00  1.245e-01   9.903  < 2e-16 ***
Region.NewVanFraser  4.573e-01  1.434e-01   3.188  0.00143 ** 
YearsGive            3.810e-02  2.419e-02   1.575  0.11521    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1552.6  on 1119  degrees of freedom
Residual deviance: 1265.3  on 1113  degrees of freedom
AIC: 1279.3

Number of Fisher Scoring iterations: 4
```

承上題，只有Region.NewVanFraser改變了: >0且顯著，代表有正影響
note. 對估計參數(estimated)的意義要有認識，千萬不能以output為主，要檢查合理性-意義上(正負號)、統計上的穩健性robust(是否容易受異質性影響)

2. Generate Prediction

predicted probabilities

```
scoreVar= predict(logCCS,dataTest)#把`logCCS`的結果運用在`dataTest`上(用七成的東西件模型，去預測15%的資料，看其狀況)
Prob = exp(scoreVar)/(exp(scoreVar) + 1)# Logistic機率的算法要指數，所以要做指數轉換`exp`
newData = data.frame(MonthGive=dataTest$MonthGive, Prob=Prob)
```

預測機率>50%就是yes，小於就是NO，和真正的機率對照就可以知道預測狀況如何

```
a = data.frame(score=scoreVar,Prob=Prob,Raw=dataTest$MonthGive)
head(a)# 預測prob vs. 原始raw
```

```
           score      Prob Raw
6   0.02028567 0.5050712 Yes
7   2.00766584 0.8815996 Yes
10  2.13178486 0.8939543 Yes
12  0.35901355 0.5888016 Yes
21  0.59573606 0.6446802 Yes
26 -0.32052961 0.4205467 Yes
```

```
newDataSorted = newData[order(newData[,2],decreasing=TRUE),]# 將資料清理並排序(按照預測大小)
```

可看出預測高的有No，預測低的有yes的錯誤

3. Confusion Matrix

使用confuse matrix評斷模型預測的準確度 要**Positive=Yes**才能接受 CFMatrix1.glm輸出真實比率

```
          Oberserved
Predicted No Yes
      No  80  25
      Yes 46  89
```

```
          Oberserved
Predicted        No       Yes
      No  0.6349206 0.2192982
      Yes 0.3650794 0.7807018
```

如果混淆矩陣計算的模型，正確性(Positive=Yes)可以接受，則這個分類模式隱含的決策如下:
右上角的值25，代表被模型預測Yes(y=1)，也就是說所有的解釋變數都預測他們是Yes，事實上卻是No(y=0)。這25人應該是行銷加碼之處。因為依照預測，他們都有成為Yes的潛力。更精確的作法：把這 244 個 Yes的預測機率取出來，取機率大於 70%的人

其他常用判斷指標

```
Confusion Matrix and Statistics

          Oberserved
Predicted No Yes
      No  80  25
      Yes 46  89

               Accuracy : 0.7042
                 95% CI : (0.6421, 0.7611)
    No Information Rate : 0.525
    P-Value [Acc > NIR] : 1.16e-08

                  Kappa : 0.412

 Mcnemar's Test P-Value : 0.01762

            Sensitivity : 0.6349
            Specificity : 0.7807
         Pos Pred Value : 0.7619
         Neg Pred Value : 0.6593
             Prevalence : 0.5250
         Detection Rate : 0.3333
   Detection Prevalence : 0.4375
      Balanced Accuracy : 0.7078

       'Positive' Class : No
```

Accuracy正確率

以總樣本看，預測 Yes/No 的正確率

敏感度：Sensitivity

真實 YES 正確被預測的比率=Recall Rate

特異度：Specificity

真實 NO 正確被預測的比率

```
TP = CFMatrix0.glm[1, 1] #True Positive
TN = CFMatrix0.glm[2, 2] #True Negative
FN = CFMatrix0.glm[2, 1]
FP = CFMatrix0.glm[1, 2]
```

precison

從預測 Positive 角度 (i.e. YES) 看，與真實 YES 相符的比重

```
precison = TP/(TP+FP)
```

recall rate

真實的 YES，被預測 YES 命中的比重

```
Recall = TP/(TP+FN)
```

F1(beta=1時)

```
F1 = 2/((1/precison)+(1/Recall))
```

Youden's J index

衡量了 S+S 聯手預測的總貢獻，故 J 越接近 1 越好

```
J = 0.6349 + 0.7807 - 1 #Sensitivity+ Specificity-1
```

Decision Tree

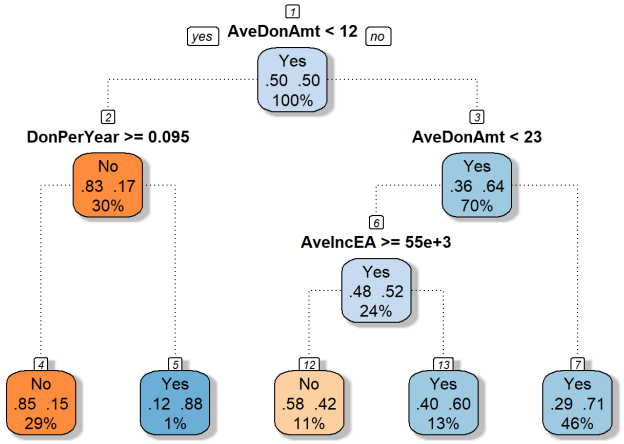
透過歸納規則將資料從樹根開始分類，一節一節尋找最佳分割點來將資料分成為小單位的集合(以下使用 `rpart()`)
決策樹由幾種元素構成：

- 根節點：包含樣本的全集
- 內部節點：對應特徵屬性測試
- 葉節點：代表決策的結果

```
library(rpart)
source("./src/trainingSamples.src")
dataset=read.csv("./data/CCS.csv")

dataset$subSample = trainingSamples(dataset, Training=0.7, Validation=0.15)# 原始資料1600被隨機切割為70%訓練資料集與15%測試資料測試

dataTrain=subset(dataset,subSample=="Training")
dataTest=subset(dataset,subSample=="Testing")# 為Output，先取出來方便等等計算錯誤率
fit.tree=rpart(MonthGive ~ DonPerYear + AveDonAmt + AveIncEA + SomeUnivP, data=dataTrain, cp=0.01)#建構決策樹，第一個逗號前為預測公式，這邊就是利用DonPerYear + AveDonAmt + AveIncEA + SomeUnivP來預測MonthGive這欄，data則填入train的資料
#cp值(complexity parameter)代表的是每一個規則（切割）所能改善模型適合度的程度。
```

cp=0.01即代表，如果該規則（切割）沒有達到至少**0.01**的模型適合度改善，則停止

```
rattle::fancyRpartPlot(fit.tree,sub=NULL,palettes=c("Greys", "Oranges")[2],type=1) #畫出剛剛建構的決策樹
```




```
Prob=predict(fit.tree,dataTest) #使用predict()將訓練好的模型套用在測試資料集上
Prob[1,]
```

```
        No       Yes
0.2887597 0.7112403
```

錯誤人數(應該是Yes預測為No的)為1600×(0.29×0.85+0.11×0.6)
由於預測結果為類別型(0,1)，所以一樣可以用Confusion Matrix評斷模型的準確度

本次模型預測準確度(Accuracy)大約為65%

File management

```
options(scipen = 999) #跳過科學記號
options(digits.secs = 6) #整數表試6位

suppressMessages(readr::read_csv("./data/flights.csv")) #去掉讀取資料時的冗贅文字

dir.create("aboutFiles", recursive=TRUE) #在檔案總管中新增資料夾(在aboutFiles中)
dir.create("./aboutfiles/p1/p2", recursive=TRUE)#recursive = TRUE是要創造一個「巢狀」的工作目錄，讓程式可以反覆使用同一路徑路徑
dir.create(file.path(".","p1","p2","p3","filename"), recursive=TRUE)
list.dirs() #目前這個工作目錄有哪些檔案

file.copy(from = "temp.R",  to = "./aboutFiles/", recursive=TRUE) #把檔名複製到to的資料夾
file.copy(from = list.files()[5:22],  to = "./aboutFiles", recursive=TRUE) #打第五到22個複製到新建的資料夾

file.rename(from = "./aboutFiles/temp.R",  to = "./aboutFiles/temp1.R") # rename files of the same directory
file.rename(from = "temp.R",  to = "./aboutFiles/temp2.R") # move and rename
file.remove("./aboutFiles/temp2.R") #delete
file.exists() #確認檔案是否存在
```

Principal component ananlysis

主要成分分析/主成分分析目的為使用較少的變數來解釋最多的變異
以data中的PCA.xlsx 舉例而言，每欄都是一個維度，主成分分析就是要降維:產生主成分指數
此指數就像是用各欄的變數作加權，並為其產生的主成分命名，以達到將多變數減少成少量變數(產生的各PCA)做分析
雖然N欄最多會有N個主成分，但是通常不會這樣做，因為這樣就違背了主成分分析想要降維的目標

```
dataset=openxlsx::read.xlsx("./data/PCA_dataTable.xlsx",sheet=3) #要EXCEL第三個工作表單(即第三個人)
labels=names(dataset)
ROWNAMES=sub(dataset[,1],pattern =" ",replacement=" ")#把第一欄公司名稱的空格取代成沒有(即消除空格)
a=4;b=5 #抓第4和5欄，即區位和工作挑戰
X=dataset[,a]
Y=dataset[,b]
newData=dataset[,-1] #去掉第1欄(公司名稱)，這樣ROW ID比較好看結構
rownames(newData)=ROWNAMES  # Establish row ID by rownames
```

決定主成分

```
.PC= princomp(newData ,cor=TRUE) #開始進行主成分分析;cor是邏輯變量，當cor=TRUE表示用樣本的相關矩陣R做主成分分析
sigma=(summary(.PC, loadings=TRUE)$sdev)^2#loadings的輸出結果為載荷是主成分對應於原始變量的系數，簡單說就是所占比重影響
varianceRatio=t(round(sigma/sum(sigma),4))#每個主成分解釋變異量
varianceRatio
```

```
     Comp.1 Comp.2 Comp.3 Comp.4
[1,] 0.6517 0.2618 0.0773 0.0092
```

```
summary(.PC, loadings=TRUE)#顯示主成分分析的結果
```

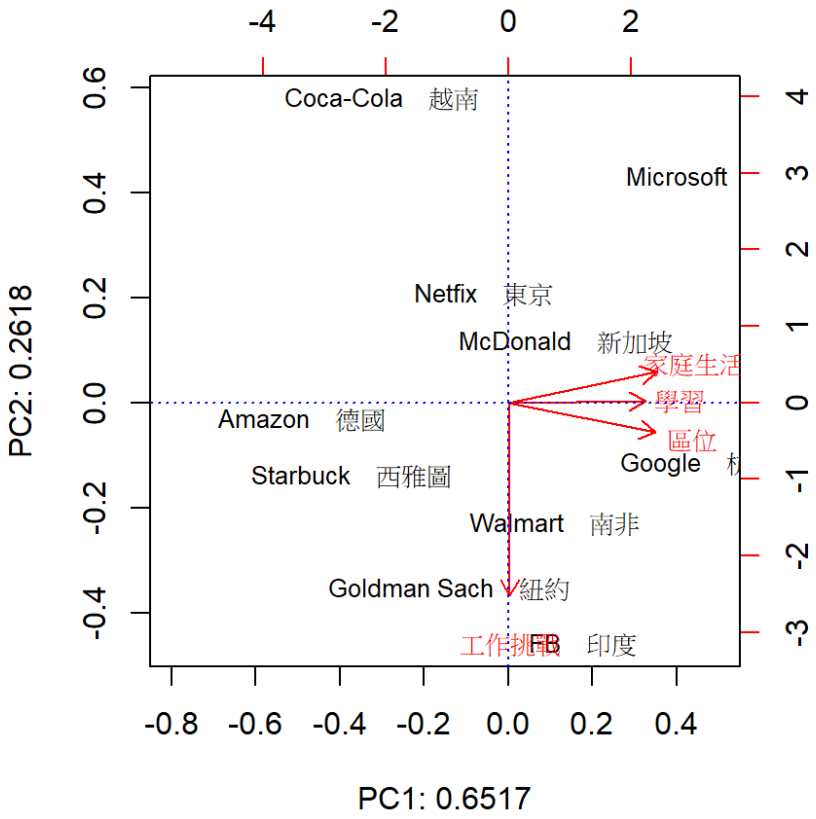
```
Importance of components:
                          Comp.1    Comp.2     Comp.3      Comp.4
Standard deviation     1.6145980 1.0232965 0.55591673 0.192078436
Proportion of Variance 0.6517317 0.2617839 0.07726085 0.009223531
Cumulative Proportion  0.6517317 0.9135156 0.99077647 1.000000000

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4
家庭生活  0.593  0.155  0.352  0.707
學習      0.549        -0.835
區位      0.589 -0.151  0.423 -0.672
工作挑戰        -0.976         0.216
```

- Standard deviation 標準差 其平方為方差=特征值
- Proportion of Variance 方差貢獻率
- Cumulative Proportion 方差累計貢獻率 可看到到累積貢獻比率(Cumulative Proportion)在第一主成分Comp.1上為0.6517317，累積到第二主成分Comp.2時為0.9135156了
  解釋變異量5中也可以看出，第三和第四主成分占比甚微(0.0773和0.0092)，代表只要用第一和第二兩個主成分就可以建構幾乎出整個樣本集合。因此，選擇PC1與PC2

根據Loadings中的 Comp.1 那一列可知，第一主成分Comp.1=0.593 x1+0.549 x2+0.589 x3，也就是我們用這個主成分就可以掌握整組資料的6成(0.6517317)，加上第二主成份C就可以掌握 0.9135156 的資料，重建整個矩陣了

主成份負荷圖

```
p=1;q=2
Xlab=paste0("PC",p,": ",varianceRatio[p])
Ylab=paste0("PC",q,": ",varianceRatio[q])
biplot(.PC,choices = c(p,q),cex=0.8,xlim=c(-0.8,0.5),xlab=Xlab,ylab=Ylab);abline(h=0,v=0,lty=3,col=("blue"))# 選取 PC1 和 PC2 繪製主成份負荷圖
```



圖中的標記的紅色向量分別表示各變數對主成分的作用的方向

用R做視覺化基礎

R畫圖工具有三種:base、lattice、ggplot2

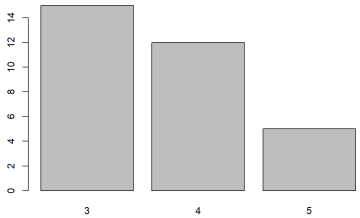
Case: mtcars

```
gear.table <- table(mtcars$gear) #使用內建數據mtcars(詳細資料可執行?mtcars)
gear.table #3、4、5轉換成table後非數字，為欄名稱
```

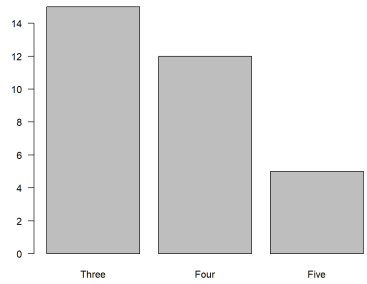
```
 3  4  5
15 12  5
```

Base R

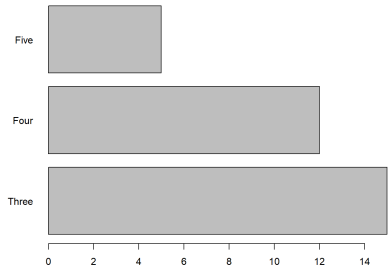
```
barplot(gear.table)
```



```
par(las = 1, mar = c(3, 5, 1, 1)) #las=1(軸的標籤label of axis) ,將three,four,five轉呈水平文字; mar=margin
barplot(gear.table, names.arg = c("Three", "Four", "Five"))#用names.arg將3、4、5改名稱為"Three", "Four", "Five"
```



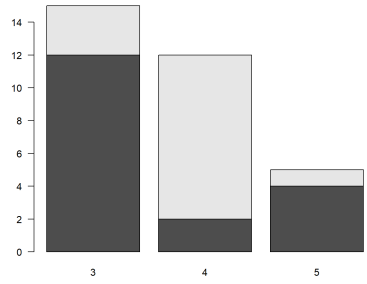

```
barplot(gear.table, names.arg = c("Three", "Four", "Five"),
        horiz = TRUE)#用horiz = TRUE 置換標籤
```



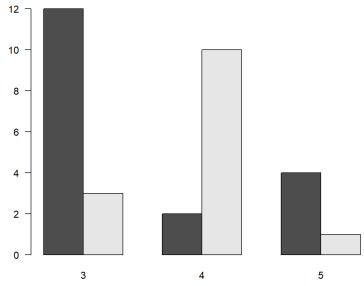

```
#若希望跳出新視窗，僅須在barplot前面加上dev.new();

gear.table2 <- table(mtcars$vs, mtcars$gear)
gear.table2# 交叉表
```

```
   3  4  5
0 12  2  4
1  3 10  1
```

```
barplot(gear.table2)#沒用beside就會疊加
```




```
barplot(gear.table2, beside = TRUE)#用beside並排
```



Latice

畫出來自動上色，內定長條圖示水平顯示

```
lattice::barchart(gear.table)

names(gear.table) <- c("Three", "Four", "Five")# 置換成three,four,five
lattice::barchart(gear.table)

lattice::barchart(gear.table2)
lattice::barchart(t(gear.table2))#t(transport轉置)列行互換，即0,1和3,4,5互換
lattice::barchart(t(gear.table2), stack = FALSE)#將內建的疊加stack設為false就可以並排了
```

ggplots

```
library(ggplot2)
ggplot(mtcars, aes(factor(gear))) +
  geom_bar()# 給檔案名稱和要畫gear資料即可(aes:aethestics美學);一定要斷行;要畫barchart就使用geom_bar


ggplot(mtcars, aes(factor(gear))) +
  geom_bar() +
  coord_flip()

ggplot(mtcars, aes(factor(gear), fill = factor(vs))) +
  geom_bar() +
  coord_flip() #用fill自動填色;用coord_flip()置換


ggplot(mtcars, aes(factor(gear), fill = factor(vs))) +
  geom_bar(position="dodge") +
  coord_flip() # position="dodge"改成並排
```

```
png(file = filename,width = 14*2.54*25, height = 4.25*2.54*25, bg = "white")# 下面這個圖形要存成Png
text(x,y+0.02,labels=as.character(y),cex=0.8);box(bty="l") # y+0.02比位置再高一點點(不一定是0.02，以原始數字*1.05最適)
dev.off()
## Using texture, instead of colors用材質非用顏色填滿
x<-barplot(as.matrix(DF2[,-1]), beside=TRUE,
           legend.text=DF2[,1],
           args.legend=list(bty="n",horiz=TRUE),
           border="white",names.arg=NULL,
           ylim=c(min(DF2[,-1])-0.1,max(DF2[,-1])*1.25),
           ylab="Annualized Value",xlab="Models",
           main=NULL,
           density=c(30,30,20,40), angle=c(135,45,0,90), col="brown")#density是線條填滿程度;angle是填色線條的角度;顏色brown
```

參考資料:

資料科學與R語言 _曾意儒 (https://yijutseng.github.io/DataScienceRBook/)
R語言 工作空間 (workspace) 及檔案 (files) (https://ithelp.ithome.com.tw/articles/10218417) 主成分分析 (https://rpubs.com/skydome20/R-Note7-PCA) R 統計軟體(7) – 主成分分析與因子分析 (作者：陳鍾誠) (http://programmermagazine.github.io/201310/htm/article3.html)