推薦序

大數據的走向朝開放原始碼(Open Source)、視覺化(Visualization)和CRISP-DM前進，軟硬體不再是問題，甚至連最簡單常見的Excel都可以處理大數據。這也是微軟對於雲端及大數據的使命。

借鑒微軟SQL Server 2016 R應用的特點，提煉方法，以R的形式體現方法。統計學早已脫離正態的傳統框架發展計數，讓SQL Server 2016 R每一位從事Big Data領域人員能善用SQL進行分析。透過SQL讓大數據概念與技術範疇簡介、資料倉儲概念、商業智慧流程和顧客關係管理（CRM）知識紮實的發揮，為大數據分析技術的結合帶來了發展的契機。特別是在大數據分析上的亮點「R」軟體，「R」軟體屬於免費開放來源（Open Source）程式設計和統計語言，近年來為統計界中相當有力的分析工具技術，SQL Sever以資料庫見長加上強大的數據分析能力，像是「R」軟體、「R」及Excel這些免費的開放式資料以及軟體，軟硬體的層面並不是問題，重點在於資料的價值性，在於想分析什麼，解決甚麼方案，換言之資料量越大，包括文字、聲音以及影像，其分析越來越快且方便。

哪裡有資料，可以做哪些整合分析，產生的價值性在哪。

當SQL Server 2016 與R雙贏的整合，推動著統計界及數據分析領域之技術發展進入嶄新的模式。謝邦昌院長的研究團隊與微軟向來在Data Mining, Big Data及AI領域密切合作，這本書的問世更是揭櫫兩造雙方邁向AI合作的決心與契機。

希望我們一起來　AI　Taiwan　愛 台灣　!!

台灣微軟 首席技術與策略長

丁維揚

1-1-4 大數據分析簡介

除了大數據技術工具外，目前已出現許多新開發的大數據分析和資料視覺化工具。以下從分析工具和演算法角度歸納出幾個重點。

分析工具大致可區分 7 種

1. **R**，屬於一種開放來源（Open Source）程式設計的統計語言，近來特別受到大學和新興公司的青睞，主要是因為免費及開放來源（Open Source）。R 語言是一種 GNU 專案，其可以一再免費散佈，因此數以千計的開發商更是不間斷地持續使用和擴充功能。關於 R 詳細資訊，讀者可至 http://cran.r-project.org/網站查詢。另外，還有一種 R-Studio 整合開發環境，關於 R-Studio 詳細資訊，可至 www.rstudio.com/ide/download/網站查詢。而本書第 8 章內容也將提及從 SQL Server 2016 中如何安裝 R 環境跟執行，並且搭配些許應用案例，讓讀者能夠瞭解 R 的應用廣度已越來越大。

2. **Python**，一種物件導向、直譯式的電腦程式語言。語法相當簡單，能輕鬆完成很多常見任務，也經常被當作腳本語言用於處理系統管理和網路應用程式的編寫。近年來大數據分析上使用 Python 來進行網路爬蟲及做為應用程式開發工具已是非常普遍現象，舉凡例如 Google、Facebook 等網站，到處都可見到 Python 蹤跡。

3. **Apache Mahout**，同樣屬於 Apache 軟體基金會的一個專案，在 Hadoop 平台頂層提供可延展的機器學習演算法。Mahout 是提供在 Apache Hadoop 上層由 MapReduce 執行的叢集（Cluster）、分類和協同篩選（Collaborative Filtering）演算法。如果讀者想進一步知道 Apache Mahout 的詳細資訊，讀者可至 http://mathout/apache.org 網站上查詢，以及瞭解 Mahout 支援的各種分析演算法。

4. **MADlib**，屬於開放來源（Open Source）程式庫，支援資料庫內分析，提供數學、統計和機器學習方法資料平行實做功能。此方法可支援結構化、準結構化和非結構化等資料來源。關於 MADlib 資訊，讀者可至 http://madlib.net/網站查詢。

5. **SPSS**，屬於統計產品與服務解決方案（Statistical Product and Service Solutions）的簡稱，是 IBM 公司推出之一系列用於統計分析運算、資料採礦、預測分析和決策支持任務的軟體產品與相關服務的總稱。

6. **SAS，即統計分析系統（Statistics Analysis System）**，最早是由北卡羅來納州立大學兩位生物統計學研究生所編寫及製定，早期只是一個數學統計軟體，可是後來 Jim Goodnight 及 John Sall 博士等人於 1976 年由成立統計分析系統公司，正式推出相關軟體。然經過多年發展已經遍佈全世界，使用單位

涵蓋金融、醫藥衛生、生產、運輸、通訊、科學研究、政府和教育等領域。在資料處理和統計分析領域中，SAS 是被譽統計軟體界的巨無霸或法拉利。

7. **Microsoft SQL Server**，其中一項服務稱做 Microsoft SQL Server Analysis Services，提供建立複雜資料採礦方案所需之功能和工具。主要區隔三個部分，（1）一組業界標準的資料採礦演算法，目前總共有 10 種演算法；（2）資料採礦設計師，可用來建立、管理和瀏覽資料採礦模型，然後再使用這些模型來建立預測；（3）延伸模組（DMX）語言，用來管理採礦模型和建立複雜的預測查詢。

分析演算法可分成 9 種

因為在開發過程當中常會有諸多的創新項目，當然先進的分析功能也是一樣，以下是近來資料科學家較常使用的演算法簡介。

1. **支援向量機（Support Vector Machines）**，是一種監督式學習方法，可廣泛地應用在統計分類及迴歸分析上。在高維特徵空間中使用線性函數並進行空間分劃區隔的學習系統，其會根據有限樣本在模型資訊的複雜性與學習能力之間找尋最佳解，以期獲得最好的分類能力（Vapnik, 1998; Vapnik and Chapelle, 1999）。基本上支援向量機在空間資料集特性分類可區分以下 3 種：線性可分支援向量機、線性不可支援向量機、非線性可分支援向量機。

2. **隨機森林（Random Forest）**，該決策樹方法首先是在 1997 年由 Amit 與 Geman 所提出，之後是由統計學家 Brieman 在 2001 年整合後形成此完整方法，其利用重抽樣（Resampling）概念建構出眾多的決策**樹，從這些樹的分類結果中選擇出現次數最多的類別做為決定結果，此方法具有高度的分類準確性。**

3. **集成式學習（Ensemble Methods）**，一種模型測試驗證技術，可用來對多個模型進行測試，以確保結果是得到一個勝過任何單一分析模型的預測效果。

4. **冠軍／挑戰者（Champion／Challenger）**，跟集成式學習（Ensemble Methods）同樣概念，是另一種模型測試驗證技術，先將目前分析模型歸入「冠軍」，再利用不同分析模型對冠軍提出挑戰。每一個「挑戰者」和冠軍各有不同的測量及定義方法，因此稱之。

5. **分類矩陣（Classification Matrix）**，用來評估統計模型的標準工具，又稱混淆矩陣（Confusion Matrix）。**主要功能是用來判斷預測值是否符合實際值**，並將模型中的所有案例分類到不同的類別目錄，每一個類別目錄（「誤判」False Positive、「真肯定」True Positive、「誤否定」False Negative 和「真否定」True Negative）中的所有案例都會計算在內，且總數皆會顯示在矩陣中。由於它是評估預測結果的重要工具，因此工具易於瞭解及說明預測錯誤影

響。不過透過檢視此矩陣之每個資料格中的數量和百分比，就可快速知道模型預測正確機率。

6. **連續小波轉換（Continuous Wavelet Transformation）**，一種用來分解一個連續時間之函數，可使它變成數個小波（Wavelet)。這跟傅立葉變換（Fourier Transform）不一樣的是，連續小波轉換（Continuous Wavelet Transformation）可建構一個具有良好時域和頻域局部化的時間頻率訊號。

7. **文字採礦（Text Mining）**，從非結構化資料探勘成結構化資訊以及擷取重要數字指標，將非結構化資料轉換成結構化資料過程，稱之。

8. **語意情感分析（Sentiment Analysis）**，繼文字採礦（Text Mining）之後，近年來被熱烈討論的議題。目的是在於對某個主題或文件的整體所抱持態度及揭露情感進行分析。大致情感分析的幾個方法如下：

   - 基於文件為基礎的情感分類（Document-based sentiment classification）
   - 以主觀概念做情感分析（Subjectivity and sentiment classification）
   - 以外觀（屬性）為基礎的情感分析（Aspect-based sentiment analysis）
   - 建立情緒關鍵詞的情感分析（Lexicon-based sentiment analysis）

   綜觀來說，情感分析（Sentiment Analysis）的商業價值，除了可提早瞭解消費者對於產品或公司觀感，還可進而調整營運策略方向。且在產品銷售途中，也可捕捉消費者對產品體驗程度回饋。

9. **特徵選擇（Feature Selection）**，這是要從原有的特徵中挑選出最佳的部分特徵，使其辨識率能夠達到最高值。當這些鑑別能力較好的特徵如果被辨識出來，不但能夠簡化分類計算，也可幫助瞭解此分類問題之因果關係。其**多用在建模過程，特別適用資料可能含有冗餘或非相關變數的情況之下。**

1-2 大數據與資料倉儲

傳統資料倉儲以「資料集中」為概念，雲端運算時代大數據強調「分散運用」。面對資料科學運用的壓力，兩者整合或跳躍成長，勢必不可避免。大數據最熱門的分散式運算架構 Hadoop，就是運用此種概念（儲存及運算共用），不僅可儲存不同來源與格式的龐大資料，也可自動進行資源調配和分散式運算。不過傳統資料倉儲的重要性依然是首要，因為若沒有已建置好的資料倉儲，哪來的大數據呢？

1-2-1 兩者差異之處

大數據，顧名思義是擷取、管理並分析大量、多種型態的資料，探索隱含在其中的關聯性。它的關鍵不在於所管理的資料量多巨大，是在於如何分析資料。談到資料分析，對於國內知名企業來說並不是一件新鮮事，他們早在多年前就已建置資料倉儲，有計畫的蒐集客戶相關資料，深入分析客戶行為，獲利與風險等重要情資後，再提供行銷、財務、風險管理等單位應用。

有此一說，現況已有資料倉儲的資料分析企業，還需要引進大數據嗎？我們都曉得大數據與資料倉儲兩者雖以資料分析為主，但是在實質內容上有所差異。簡單來說，大數據跟資料倉儲有 3 個不同之處。

1. **巨量**：大數據的資料數量一定要夠多，才能有機會從中找出「有序」，正所謂「亂中有序」；但資料倉儲則需適度控制其數量，否則需要冗長計算時間才能有分析結果。
2. **多樣**：大數據資料型態包羅萬象，有文字、語音、影像、日誌、地理位置、甚或感應器信號等多種類型的非結構化資料。由於近來外部資料蓬勃發展，各企業也開始在蒐集外部資料，因此資料倉儲所蒐集的資料多以結構化資料為主（例如顧客消費行為資料）；大數據相較於資料倉儲在資料結構上幾乎是雜又亂（Messy），可是對分析主體的涵蓋面上卻是較為完整。
3. **結果**：大數據最終目的是要挖掘出隱含在雜亂資料中的相關性，再從此相關性來推論未來，所強調的就是預測；資料倉儲是對過去的歷史做個統計，以精準的計算各項指標，強調的是對現況瞭解。兩者雖都是分析，但因方法上的差異，所得出的結果大不相同。

由於大數據分析能處理大量樣本與多元型態資料，透過交叉分析，可以淡化資料雜訊對真實訊號的影響，分離出資料與資料之間真正的關聯性以進行推論，**因此在執行「預測」這個任務上，大數據做得非常成功，這也是大數據不同於資料倉儲之處**，同時也是大數據能吸引上百家全球公司競相投入的重要誘因。

1-2-2 兩者相同之處

目前已經很多人採用大數據技術來處理和儲存鬆散結構化數據，例如氣象數據、自然資源數據、媒體業者新聞資料的分析、民調數據分析、預測大自然變化、消費者購物行為等；政府亦可採用大數據技術來瞭解民意趨勢和監測網路安全。薪資數據、報稅資料、犯罪紀錄、投資交易資料、銷售紀錄、顧客購物紀錄等，這些屬於獨立信息系統的範疇仍然多採用資料倉儲的技術來處理。

相同處，是從各種數據中找出線索、趨勢，以及商機。從早期的醫學專家系統（Medical Expert System）、模糊邏輯（Fuzzy Logic）到大數據和資料倉儲都有一個相似的分析模式，就是尋找統計資料的因果關係和相關性。同時政府部門希望瞭解民眾需要以提供更好服務，企業商家希望瞭解消費者的購物紀錄來預測顧客未來購物行為以提供更精準的行銷。

1-3 大數據應用案例

兩個著名的例子是「啤酒與尿布」及「颶風和草莓夾心酥」，美國知名零售商分析過去數十年消費者的購物行為，從結果中找到高度相關資料，這就是購物籃分析（俗稱菜籃分析）。資料採礦（Data Mining）這塊領域在過去十年裡已被成功運用在各領域上，同時也替許多企業貢獻不少業績。

著名企業 Google、IBM 和 Amazon 等這些早已經擁有大量資料的公司，它們都已發展出不下無數的商業模式（公式），更有不少企業搶先一步，藉由這些資料進行分析，進而達到業績目標與創造新商機。關於大數據的案例，其實也可以用在一般人尚未瞭解的地方，以下就幾個著名案例做介紹。

1-3-1 IBM 推出華生機器人來看診

著名企業 IBM 繼 1997 年推出挑戰棋王的超級電腦「深藍」後，又再推出更聰明且會聽人話的超級電腦「華生（Watson）」。如今 IBM 和美國最大健保公司 WellPoint 合作，目的是藉由「華生（Watson）」的語音辨識和迅速的資料採礦（Data Mining）技術，來協助醫護人員問診及診斷惡性腫瘤。華生（Watson）之所以能夠看診，在於運用大量的臨床病例，可在極短時間內分析所有可能結果，並協助醫生提出「治療建議」，大幅減少過程中產生的疏忽。

華生（Watson）電腦的推手之一－IBM 美國加州艾曼登研究中心資深研究員史班格勒（W. Scott Spangler），他指出華生（Watson）電腦可運用智慧、邏輯化方式，幫助企業在現今擁有大量資料的環境下迅速做出判斷、分析、解讀各種訊息，未來絕對會成為企業執行決策的好幫手。而且他認為華生（Watson）電腦目前在四大領域（醫療、科技、企業知識管理及政府應用）中是很有發揮潛力地。

1-3-2 汽車防盜系統使用臀部辨識技術

臀部辨識技術？沒錯這是一種新的身分辨識方法，它是由日本的產業技術學院（Advanced Institute of Industrial Technology）所開發出來的，藉由分析座椅在人們乘坐時所承受的壓力，辨識乘坐者身份。該方法通常應用在汽車防盜或電腦身份辨

識等領域，辨識率相當可靠。臀部辨識，係利用安裝在座位下方的 360 個感應器偵測壓力，能夠精確測量臀部特徵，例如：輪廓及其對椅子施加的壓力等，透過感應器偵測值的範圍由 0 至 256，可經由與感應器連接的電腦顯示 3D 圖形，此偵測值將成為個人化之辨識密碼（數據代碼）。

因此研究人員正將把這項技術與汽車業者合作，使其能應用於汽車防盜系統，並計畫於未來 2 至 3 年內實現商業化。倘若有人強行進入具有這種防盜功能的車內，而座位壓力分析結果與原車主不同的話，汽車將無法開動。這項實驗辨識率高達 98%。

當然除了防盜系統之外，亦可應用於電腦身份辨識，像是用戶進入辦公室坐下後，就能夠同時登入電腦；對比傳統生物辨識技術，像是虹膜掃描和指紋識別等，就會讓受測者感到緊張，然而這種新座椅辯識技術只要簡單坐下即可，不會給人造成心理負擔。

1-3-3 Amazon 及 Netflix 網路消費

Amazon 會根據消費者瀏覽的商品跟你說曾經瀏覽過這類商品的人又瀏覽了什麼，或者是買過這個商品的人也會購買什麼商品，進而有一份推薦清單。我想在台灣大多數人的網購經驗都有來自 Yahoo 或 PChome，但如果您曾在 Amazon 購物過的話，相信其經驗絕對截然不同，一開始我們一定會看到一些毫無章法的推薦，進而會提到上述開頭的內容，其實這種推薦方式是根據歷史購買紀錄所計算出來的收斂結果。有一項很重要的訊息，根據統計這種推薦方式讓 Amazon 在一秒鐘能夠賣出 79.2 樣商品！

Netflix，是美國最大線上影音出租服務網站。著名的是每 10 部它所推薦的影片大約有 7.5 部以上是消費者會選擇接受這樣的推薦，機率可謂非常之高。更厲害的是，消費者可針對片子給幾顆星不等評價，在下完評價之前，它就已經對您預測說您上下不會超過半顆誤差。該計算都是根據您長期收視這些片子喜好（包括導演、明星的組合等）後，利用資料採礦（Data Mining）技術或機器學習（Machine Learning）進行行為分析萃取出來的。

1-3-4 歐巴馬也靠大數據

資料採礦（Data Mining）在歐巴馬競選中扮演關鍵角色，數據分析團隊成功為競選籌募到 10 億美元資金。當年選舉期間的春天歐巴馬陣營的數據分析團隊注意到，影星喬治克隆尼（George Clooney）對美國西海岸 40 歲至 49 歲的女性具有非常大吸引力。她們是最有可能為了在好萊塢與喬治克隆尼（George Clooney）、歐巴馬共進晚餐而自掏腰包的族群。最終喬治克隆尼（George Clooney）在自家豪宅

舉辦的籌款宴會上，成功替歐巴馬籌集到數百萬美元競選資金。當然，不只在西岸競選團隊同樣希望東海岸也能如法炮製「喬治克隆尼（George Clooney）效應」的成功經驗。最後數據分析團隊把箭頭指向了莎拉・傑西卡帕克，於是一場在莎拉・傑西卡·帕克的紐約 West Village 豪宅與歐巴馬共進晚餐的募款競爭便誕生了。

對普通民眾來說，他們根本不知道這次活動的想法是源自於歐巴馬數據分析團隊對莎拉・傑西卡帕克粉絲研究的重大發現。他們知道這些粉絲喜歡競賽、小型宴會和名人等重要訊息。競選主管在此次選戰中打造了一個規模五倍，相當於 2008 年競選時的數據分析部門，是由幾十個人組成的分析團隊，他們的具體工作都被嚴格保密，關於這個團隊的諸多細節是不會對外透露的，正因為歐巴馬競選陣營堅持守著自認為比羅姆尼競選陣營有優勢之處，就是資料（Data）。後來該技術也被用來預測選情，針對各州勝出的可能性，分配適當的資源，最終歐巴馬也獲得連任。

1-3-5 平價連鎖零售商－猜妳懷孕了

美國**平價連鎖零售業商場－Target，從女性**消費族群的購買行為當中，研發出一套領先同業的「懷孕預測模型」。資料分析專家發現，當某些女性從購買有香味乳液，轉而購買無香味乳液，或開始採購葉酸、鈣片、鎂與鋅等營養補充品，就會大膽推測這名女性可能已經懷孕。**Target** 專家已經將過去女性消費族群的資料進行串流、分析，研發出懷孕預測模型。而這個模型會列出 25 種孕婦最有可能購買的產品，並根據女性消費者行為，計算出她們的懷孕預測分數。一旦發現這名女性消費者已經有可能懷孕的訊息，就立刻寄出相關商品的促銷廣告。不過在台灣，根據從事電子商務專家表示，真正目前應用 Big Data 在電子商務的實際例子還是稀有動物，購物網站所推薦的東西還停留在根據公司促銷策略做推薦，沒有把消費者的消費模式、瀏覽紀錄及個人資料做完整個人化分析後再做推薦行銷。

從以上這些著名案例來看，相較大數據運用在國外已遍及各個產業。反觀台灣企業隨著數位媒體發展日益成熟，許多廣告主對精準媒體需求越來越強烈；當數據能被精準解讀時，就能幫助廣告主瞭解消費者需求，大幅縮短商品與消費者間距離，這些應用如何結合商品，未來將會在台灣數位媒體市場中引起話題與關注。

6-1 何謂分析型 SQL

網路普及速度之快與科技進步之賜，大數據（Big Data）熱潮越燒越烈，許多新的資料處理分析和管理技術因應而生，這對從事資料分析人員來說，是一項契機，也是不小的挑戰，因此要在大數據（Big Data）時代的市場中佔有一席之地，**筆者建議有一項技能是一定要具備的，就是撰寫 SQL 指令的能力**。

何謂分析型 SQL 呢？筆者定義為：「它是屬於資料分析範疇常使用的 SQL 指令，通常是用來建立資料分析指標與分析過程工具之用」。

我們都曉得如何摸索出資料所要表達的意涵、提煉出「數據精華」，因此像資料分析領域人員就可能已具備「機器學習（Machine Learning）」與「資料採礦（Data Mining）」等知識，再加上本身主修統計與資料分析技術等更是根本中的根本，而數學跟統計學就是基礎中的基礎，如何將這些知識呈現出來呢？鑽研學習市面上的統計分析軟體及程式語言，像是 R、SAS、Matlab、SPSS、Stata 等，我想若具備了以上這些技能，相信對於駕馭資料分析技術領域應不成問題。

可是資料基本上都是儲存於資料庫中，或許透過一些統計程式語言就能查詢想要的資料。可是 SQL 歷經了四十多年的考驗仍然在現代處於屹立不搖之位，雖然隨著大數據的關係，NoSQL 的出現有了一些影響，但其實 SQL 仍主導著市場，並在大數據領域贏得許多投資和部署。像是 Cloudera 推出了即時查詢開源工具 Impala，它是一款用來跑在 Hadoop 架構上的互動 SQL 查詢引擎，所以我們在看這些工具發展之下，SQL 在大數據領域中依然是歷久不衰。

如何透過分析型 SQL 來達到建立資料分析指標目的呢？建議使用一些基本 SQL 指令和彙總函數來搭配。如下是筆者彙整出幾個常會使用的分析面向與思維。

1. **擅長時間區間變化**：執行資料分析專案時，尤其是在建立資料採礦模型，常會大量使用跟時間區間變化有關係的指標（或稱變數），例如近 30、60、90 天的消費金額貢獻，近 6 個月、3 個月、1 個月的某指標收斂率等。因此時間區間的切割整併使用，對於分析指標來說是相當重要的一項技術。
2. **擅長使用分析函數**：SQL 指令已存在些許關於分析或統計函數，這點如果能夠善加利用，對於達到養成分析型 SQL 的熟稔度是有大大的幫助。
3. **清楚的邏輯思緒**：具備優異的邏輯思考是成為資料分析人才最重要的元素之一，因此分析指標的建立計算公式，一定得具備清楚邏輯思緒，這樣才能快速透過 SQL 指令轉換出來。

6-2 分析型 SQL 語法範例

在上述內容大致說明分析型 SQL 概念及定義。一般來說，使用 SQL 指令建立分析指標，基本上就已符合分析型 SQL 精神。在進入下一章（會員消費行為分析）之前，本章內容提供一般資料分析常會使用的幾個分析指標範例，供讀者可先行利用 SQL 指令來進行練習轉換。

1. **（產品）類項的總銷售金額：**可由訂單數量乘上訂單金額後，再分別對產品類項進行交叉分析（Group By）即可。
2. **（產品）類項佔總銷售金額的比例（或佔比）**：由上例進行變化而來，可知道各產品類項的銷售金額比例。
3. **排序（名）**：例如透過排名可以快速知道產品類別彼此之間的銷售情況。
4. **（產品）類項數量與銷售季節計算**：除銷售季節須透過時間區間變化指令之外，產品數量則是透過產品類項進行交叉分析。
5. 特定假期的銷售狀況分析。
6. 當月累計／季累計／年累計的銷售狀況。
7. **同期銷售狀況分析**：例如今年和去年的同期比較。
8. **近期銷售狀況分析**：例如近１週、近２週、近１個月、近３個月等。

6-3 SQL 基本應用分析語法

在 Microsoft SQL Server 環境中，存取資料庫物件（指資料表）的過程都是用 4 個節點方式來指定物件（指資料表）的名稱，彼此之間用點號（．）來做區隔，分別依序：

執行個體名稱．資料庫名稱．結構描述名稱．物件名稱

若搭配 SQL Server Management Studio 介面工具來解釋的話，請參考圖 6-1。

圖 6-1 SQL Server Management Studio 圖形介面

對於這 4 個節點名稱的使用方式，筆者整理幾種變化如下：

1. 查詢使用到跨執行個體或本機（Local）的物件（資料表或檢視）時可使用以下指令。

```
--指定4個節點名稱
SELECT * FROM [執行個體名稱].[資料庫].[結構描述].[資料表｜檢視]

--例如
SELECT * FROM [EDISON-PC].[邦邦量販店].[dbo].[GMC_Profile]
```

2. 查詢相同執行個體之下，不同資料庫的物件（資料表或檢視）時可使用以下指令。

```
--可省略執行個體名稱
SELECT * FROM [資料庫].[結構描述].[資料表｜檢視]

--例如
SELECT * FROM [邦邦量販店].[dbo].[GMC_Profile]
```

3. 查詢相同資料庫之下，不同結構描述的資料庫物件（資料表或檢視）時可使用以下指令。

```
--可省略執行個體名稱和資料庫名稱
SELECT * FROM [結構描述].[資料表｜檢視]

--例如
SELECT * FROM [dbo].[GMC_Profile]
```

4. 查詢預設結構描述的物件（資料表或檢視）時可使用以下指令。

```
--可省略執行個體名稱、資料庫名稱及結構描述名稱
SELECT * FROM [資料表｜檢視]

--例如
SELECT * FROM [GMC_Profile]
```

5. 若遇到保留字、關鍵字或物件（資料表或檢視）名稱有空白時，在 T-SQL 指令的要求需要使用雙引號或中括號，將這些情形特別標示，以避免造成執行上的問題。

```
--資料表名稱有空白時，建議請使用中括號
--例如
SELECT * FROM [GMC  Profile]

--資料表名稱有空白時，使用雙引號需搭配QUOTED_IDENTIFIER ON。
--例如
SET QUOTED_IDENTIFIER ON
SELECT * FROM  "GMC  Profile"
```

6-3-1 基本功夫－查詢資料

分析的來源通常為資料倉儲的資料，且這些資料都已經過相當程度的整理、轉換和萃取，因此接下來闡述的內容為分析過程中可能常會用到的指令或函數。

基本查詢陳述式（指令）

查詢資料庫的資料是身為**一個分析人員應具備的基本技能**，而查詢動作就是利用 SELECT 陳述式。**過程多半是針對資料表進行查詢**，也會有**檢視(View)**或**使用者自訂函數等**。瞭解上述查詢資料概念後，先來認識基本的 SELECT 查詢指令結構。

```
--基本查詢指令結構
  SELECT 輸出欄位名稱
  [ INTO 新資料表名稱 ]
  FROM [ Table_list｜View_list ]
  [ WHERE 篩選條件 ]
  [ GROUP BY 群組化欄位名稱 ]
  [ HAVING 篩選條件 ]
  [ ORDER BY 排序欄位 [ ASC｜DESC ] ]
```

首先說明一下查詢指令結構，SELECT 指令後面要銜接的是輸出欄位名稱，而輸出來源就是 FROM 後面的 [Table_list｜View_list]；若要過濾篩選資料列時，可從 WHERE 的篩選條件來設定；最後針對資料進行群組化分析時，可利用 GROUP BY 指令；也可針對群組化之後的資料進行篩選過濾，這時就需要藉由 HAVING 陳述式輔助；針對最後查詢的輸出結果指定排序時，可由 ORDER BY 指令，ASC 表示遞增（由小至大，預設型態亦可省略），DESC（由大至小）。以下陸續說明在分析時，會常使用到哪些基本的 T-SQL 指令。

SELECT 的格式（1）－選擇（或篩選）

SELECT 是 T-SQL 指令中使用最頻繁的陳述式，其意為「選擇」，可從一個到數個資料表中選擇符合條件的資料行和記錄，**其傳回結果稱為資料集（Dataset）或稱結果集（Recordset）**。SELECT 的基本格式（1）如下所述：

```
SELECT 輸出欄位名稱
FROM [資料表｜檢視]
```

在 SELECT 敘述中，會指定資料表的欄位名稱。倘若要指定複數個欄位名稱時，須以逗號（，）隔開；如果要選擇資料表的全部欄位，可用星號（*）表示；FROM 敘述是用來指定資料表（或檢視）名稱。

► 指令

```
USE [邦邦量販店]
GO

SELECT [MemberID] FROM [dbo].[GMC_Profile]
GO


--結果
(81035個資料列受到影響)
```

► 結果

結果 訊息

	MemberID
1	DM000001
2	DM000002
3	DM000003
4	DM000004
5	DM000005
6	DM000006
7	DM000007
8	DM000008
9	DM000009
10	DM000010

8-1 R Services 概述與基礎統計

SQL Server R 服務（Services），它是一個提供開發及部署新發現結果的商業智慧平台。使用者可利用豐富且強大的 R 語言（Script）和許多開放（Open Source）模組來建立統計模型，並儲存在 SQL Server 進行應用。

因為 R 服務具備強大的整合功能，可保留和資料密切相關的分析，避免因移動資料而產生多餘成本與影響安全性。另外，它支援完整開放原始碼 R 語言（Script）、SQL Server 工具和技術，能提供優異效能、安全性、可靠性和管理能力。使用者可利用便利且熟悉的工具，進行 R 解決方案的部署，特別是生產應用程式可使用 Transact-SQL 呼叫 R 語言（Script）執行階段並擷取預測和視覺效果。同時取得 ScaleR 程式庫，用以改善規模和效能。

關於 SQL Server R 服務（Services）有提供伺服器和用戶端相關元件使用，介紹如下。

R 服務（資料庫）

可安裝該項功能在 SQL Server 程式中，用以在 SQL Server 中安全地執行 R 指令碼。當選取這項功能時，延伸模組會在資料庫引擎支援執行 R 指令碼；並於建立新服務時，信任 SQL Server Launchpad、管理 R 執行階段間的通訊和 SQL Server 執行個體。

Microsoft R Client

它是一個免費工具，可以讓資料科學家在工作站實際執行資料分析時，使用 SQL Server 開發 R 解決方案。後續把結果部署在 SQL Server 的 R Services 並執行，或部署在 Windows、Teradata 或 Hadoop 的 Microsoft R Server 並執行。倘若使用 ScaleR 函式時，可指定更佳的延展性在 SQL Server 電腦上執行計算。Microsoft R Client 可當做個別、可用的安裝程式。

Microsoft R Server（獨立）

當安裝在資料庫伺服器或用戶端時，就不只是基本的 R 程式庫，可是如果為增強的連接性和效能程式庫時，例如 ScaleR，可使用相同程式碼在電腦上，並在伺服器上取得更大效能和延展性。開放原始碼 R，結合了支援平行處理及其他效能改進的專屬套件。R Services（資料庫內）及 Microsoft R Server（獨立）都包含基礎 R 執行階段與套件模組，還有能加強連線能力及效能的 ScaleR 程式庫。

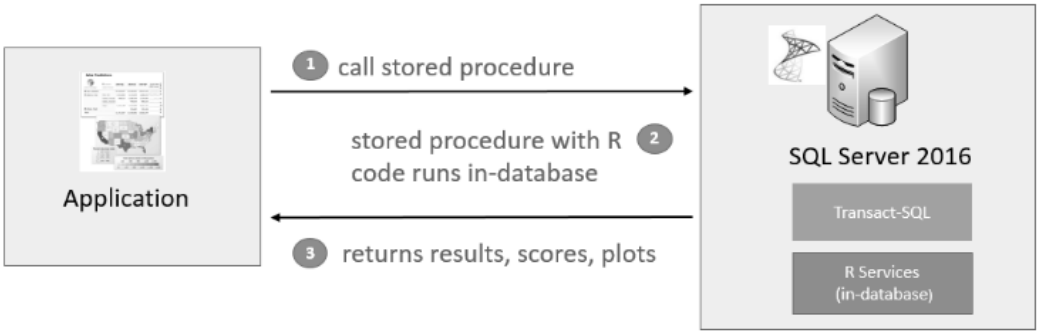

圖8-1 SQL Server With R功能架構流程

8-1-1 安裝 R Services

SQL Server 自從 2016 版本開始支援 R 語言（Script）及服務，整體來說就是在執行資料分析過程少去了 ETL 作業（指在資料庫中進行資料清理和轉換等，匯出至 R 進行統計分析），可直接利用程式（SP）把關聯式資料庫的資料直接呼叫至資料庫介面查詢和統計分析。

若讀者想瞭解此部分，須先從安裝 SQL Server 與 R 的整合開始。以往若為舊版 SQL Server 在使用 R 時，須透過 R-ODBC 的 Library 來連結（Link）；但 SQL Server 2016 可直接使用 Stored Procedure 來做呼叫。

關於 R 的執行環境可區分為 4 種，如下所述：

1. R For Windows（R Studio）
2. Microsoft R Open（RTVS R Tool Visual Studio）
3. SQL Server R Services（In-Database R）
4. Microsoft R Server

安裝 SQL Server R Services，請在安裝 SQL Server 2016 時，於「進階分析延伸模組」將「R Services」與「R Server」勾選起來。

「SQL Server 2016 R Services 安裝說明」請參閱附錄 B。(附錄 B 為電子書，請至 http://books.gotop.com.tw/download/AED003400 下載)

8-1-2 R 基礎統計分析範例

為了讓讀者一開始熟悉環境和貼近實務分析。接下來將在 SQL Server 管理工具內實際執行幾個關於 R Script 的簡單範例。

✪ 案例一：加法（SUM）

► 指令

```
--加法(從1加到1000)
EXECUTE sp_execute_external_script
@language = N'R',
@script = N'TTLSUM <- sum(1:1000);
OutputDataSet <- data.frame(TTLSUM);',
@input_data_1 = N''
WITH RESULT SETS (([總和] int NOT NULL));


--結果
(1個資料列受到影響)
```

► 結果

✪ 案例二：次方（Square）

► 指令

```
--次方(2的10次方)
EXECUTE sp_execute_external_script
@language = N'R',
@script = N'TTLSUM <- 2^10;
OutputDataSet <- data.frame(TTLSUM);',
@input_data_1 = N''
WITH RESULT SETS (([總和] INT NOT NULL));


--結果
(1個資料列受到影響)
```

► 結果