本文章主旨:

目前這一篇文字說明檔，其主要的存在目的：就是說明所在目錄「NewRPA-Automationdevelopment\_SSASIntegrateWithPowerQuery」裡面所存放的成品(目前所能擁有的功能列表)→本目錄的開發是循序漸進由一個人獨力進行，也因此是分段落逐一完成的→所以接下來的文章，其主要內容就介紹這些不同階段的開發目標與其開發成果→還有因為開發靈感出自觀賞線上教學分享視頻而生，必須交代出處以避免被人誤會抄襲!

第一階段成品(位在「PowerQueryManualPrototype」目錄中)所具備核心功能：

用手動操作Excel Power Query相關UI操作介面，呼叫後端資料庫SSAS領域相關服務，將資料過濾條件傳送到SSAS環境Cube類型元件→因而後端資料庫會透過OLEDB資料通訊協定，自Cube取回來的MDX Query顯示於Excel活頁簿上→總而言之，第一階段產生一個用手動製作出而生一個簡易功能完備的雛型草稿!

第一階段靈感出處說明:

(1)來自教學影片部分

影片標題: Power Query Load data to excel workbook from SSAS database

影片連結: <https://www.youtube.com/watch?v=vItZd5NemSw>

影片內容大意描述:是一段3分鐘長度的Step By Step Excel Power-Query的一系列操作畫面→整段操作解說不長，是把講解核心集中於以下目錄PowerQueryManualPrototype這五張圖片集上「CreatePowerQueryManualPrototype\_Step1-3.png、CreatePowerQueryManualPrototype\_Step4-6.png、CreatePowerQueryManualPrototype\_Step7-9.png、CreatePowerQueryManualPrototype\_Step10-12.png、CreatePowerQueryManualPrototype\_Step13-15.png」→表面上看來只是15個操作動作，實際上是一連串MDX Query條件過濾設定(就是以下單一三部曲操作模式[展開→點選→收攏]的一個大型loop→非專業SQA(軟體開發測試)人員出身，一般程式開發或MIS都會覺得前述這15個條件設定操作，只要光看說明圖片就帶給人繁雜且瑣碎的感覺→這也是為何要拿這一個「Power Query Load data to excel workbook」這一些複雜操作，值得當作作為一個RPA（Robotic Process Automation）模組的核心功能之價值所在!

(2)來自Power Query自帶功能與以往VBA開發經驗的累積

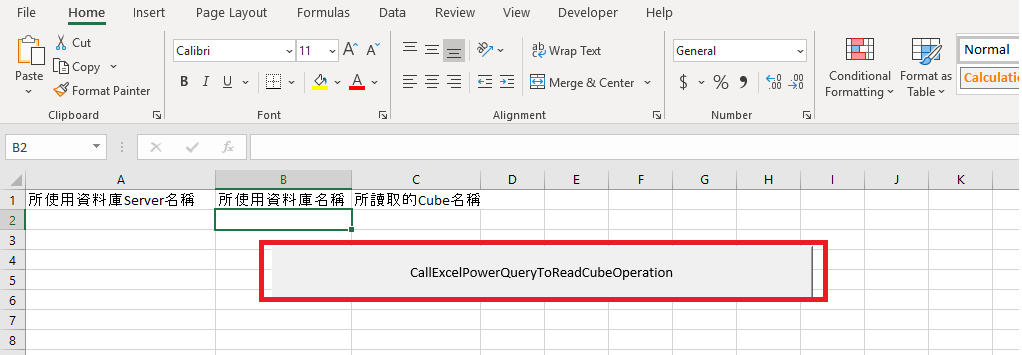
接下來以下目錄「PowerQueryManualPrototype」以下兩張圖片「CreatePowerQueryManualPrototype\_Step16-18.png、CreatePowerQueryManualPrototype\_Step19-21.png」，則是以滑鼠點選前面Power Query操作所產生成品本身內建快速篩選，來產生想要的初步結果資料集→最後這兩張圖片，則是依據累積個人VBA自動化開發經驗，去使用樞紐分析表相關資料加總功能，去將前面Power Query所產生結果資料集轉化對企業管理行銷領域比較有用加總類型營業成效報表!

第一階段成品內容說明:

接下來則是介紹儲放第一階段成品的以下目錄「PowerQueryManualPrototype」相關內容簡介→前面文章內容，已經將目錄「PowerQueryManualPrototype」9 張PNG格式圖片講解完畢，因此就不需重複提到這些內容了→最後則是描述一下該目錄最後一個檔案「PowerQueryManualPrototype.xlsx」，就是經過前面9張圖片操作產生的手工作業成品!

第二階段VBA RPA單鍵按下即自動化跑完產生成品的設計理念剖析:

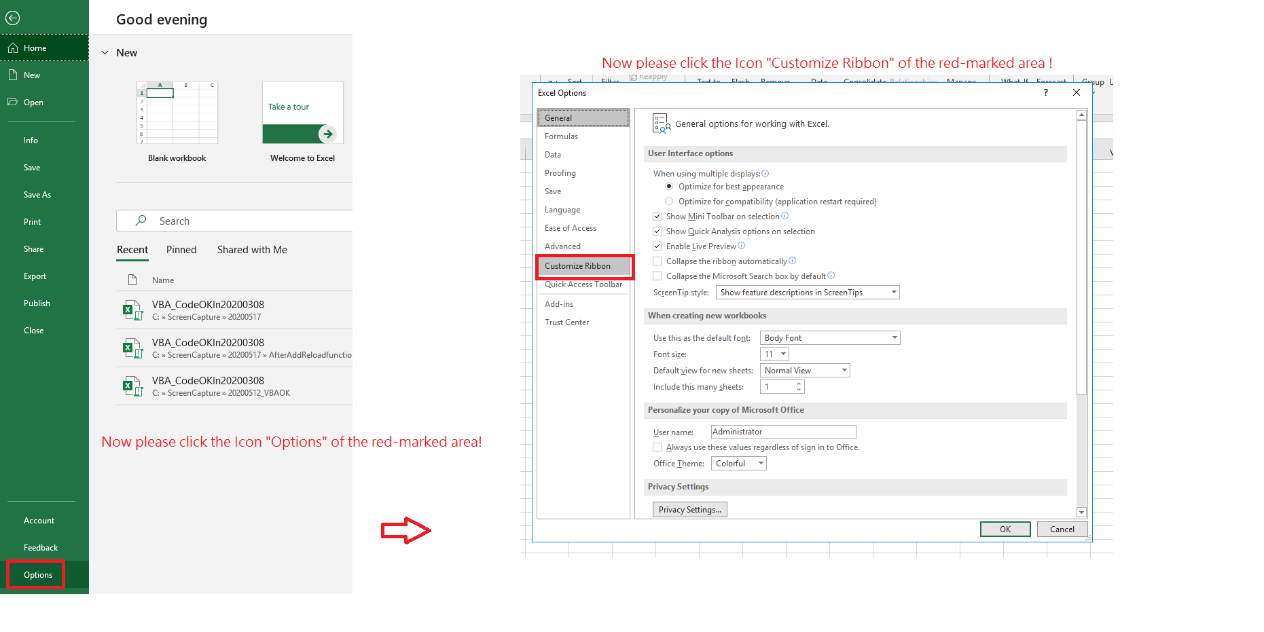
接下來則是介紹第二階段成品的以下目錄「VBA\_AutomationPortfolio\_AboutPowerQuery」的以下VBA可執行自動化RPA資料處理模組「Runable\_ExcelRPAPortfolio.xlsm」，如下圖所示

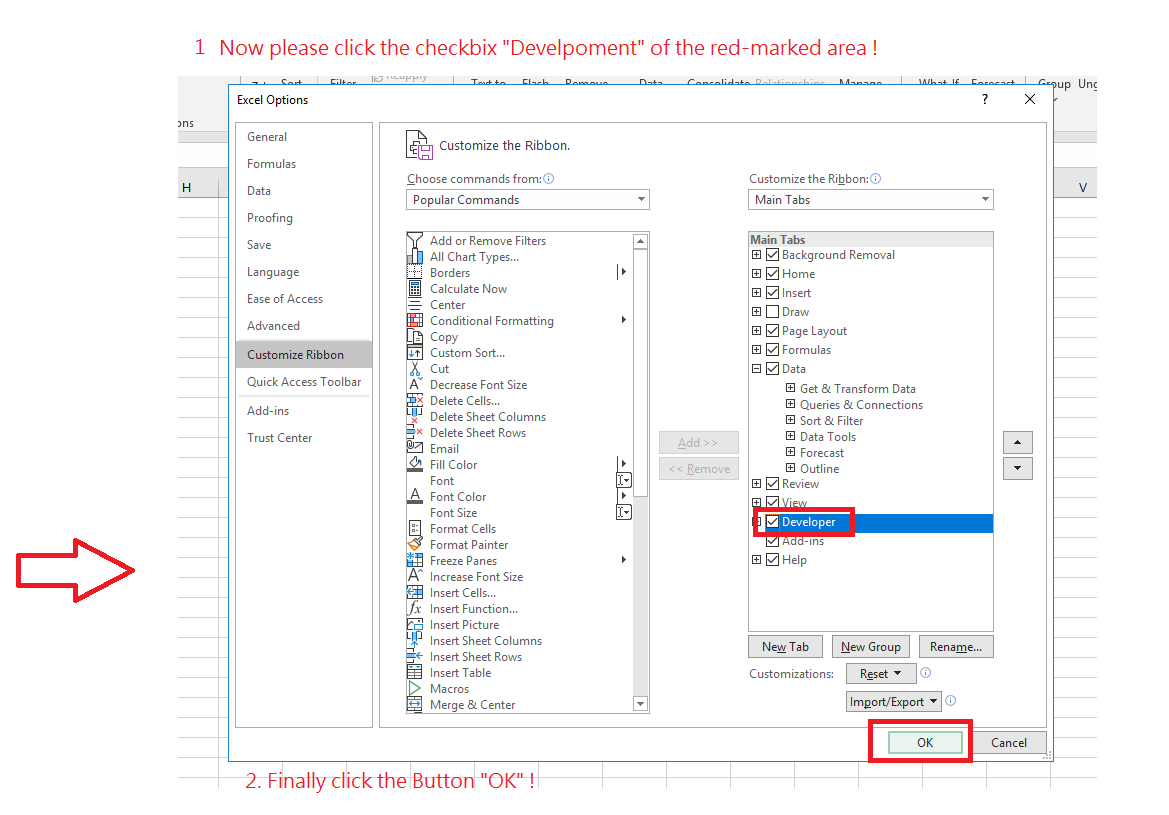


因此當使用者按下以上圖片紅色框，就觸發「CallExcelPowerQueryToReadCubeOperation」這顆功能鍵背後的一連串VBA程式碼處理邏輯，茲簡單描述如下:

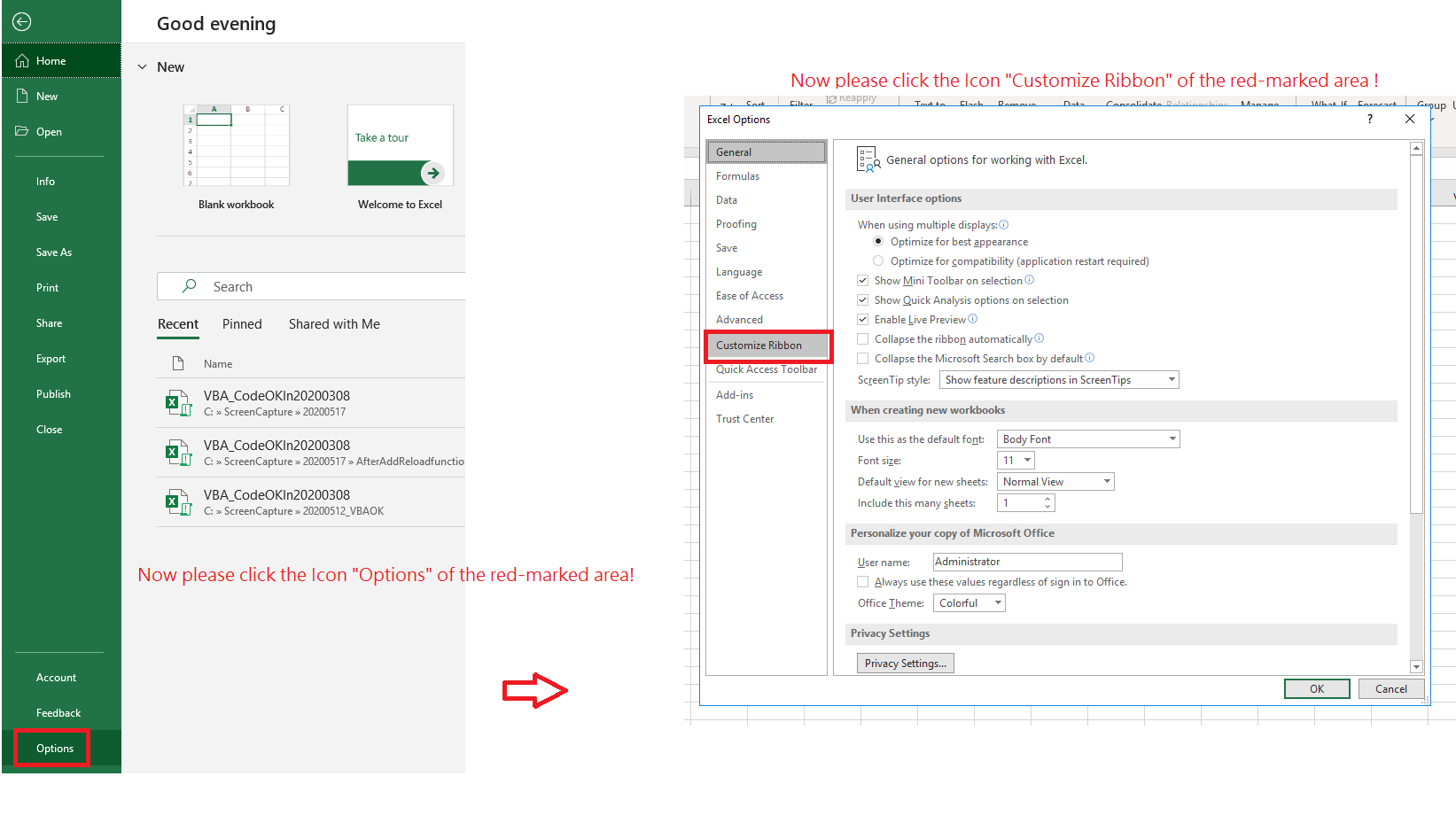
1. 圖片檔案『ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_4-5.png』防呆容錯裝置(為了防止「CallExcelPowerQueryToReadCubeOperation」按鍵被連續按了兩下，也就是保證該按鈕永遠只能被按一次，然後就按照事先錄製好商業處理邏輯，由VBA程式碼將複雜條件過濾設定產生成MDX (多維度資料運算式)Query並且傳送給後端SQL Server SSAS Cube→最後再由Excel Power Query 與後端SQL Server Cube聯繫取回資料，因而才能夠將查詢結果顯示活頁簿上)→總而言之，就是運用VBA指令溝通來達成不同現成軟體的應用整合(Power Query與 SQL Server SSAS Cube)，形成一個RPA全自動化成品(單一鍵按下後就出現一台資料打包機器人，將25個設定步驟一一執行完畢後就產生一份動態擁有不錯擴充空間互動性好的樞紐分析成品報表)→接下來用圖片展示VBA程式碼，輔助以文字說明，將這些VBA處理邏輯說明清楚 →隨後以下圖片圖一，依序操作讓Excel 隱藏好內建2016 VBA程式碼開發環境啟用起來:

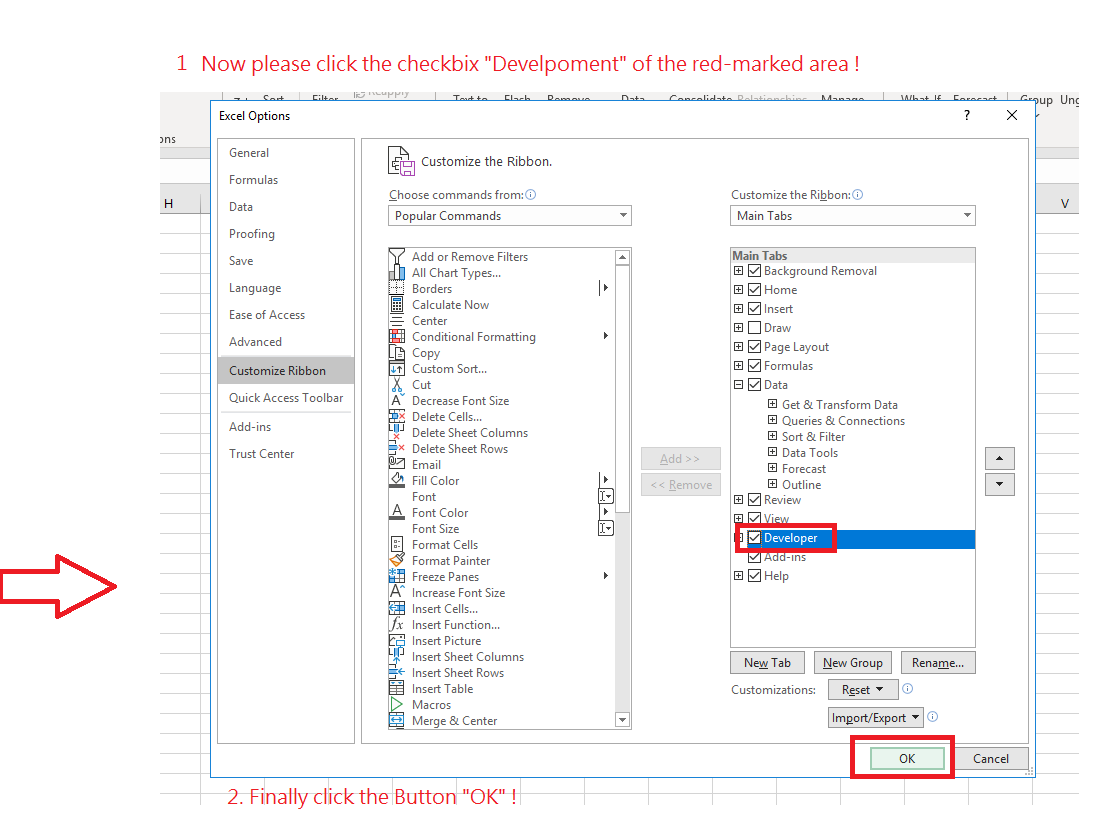
圖一、



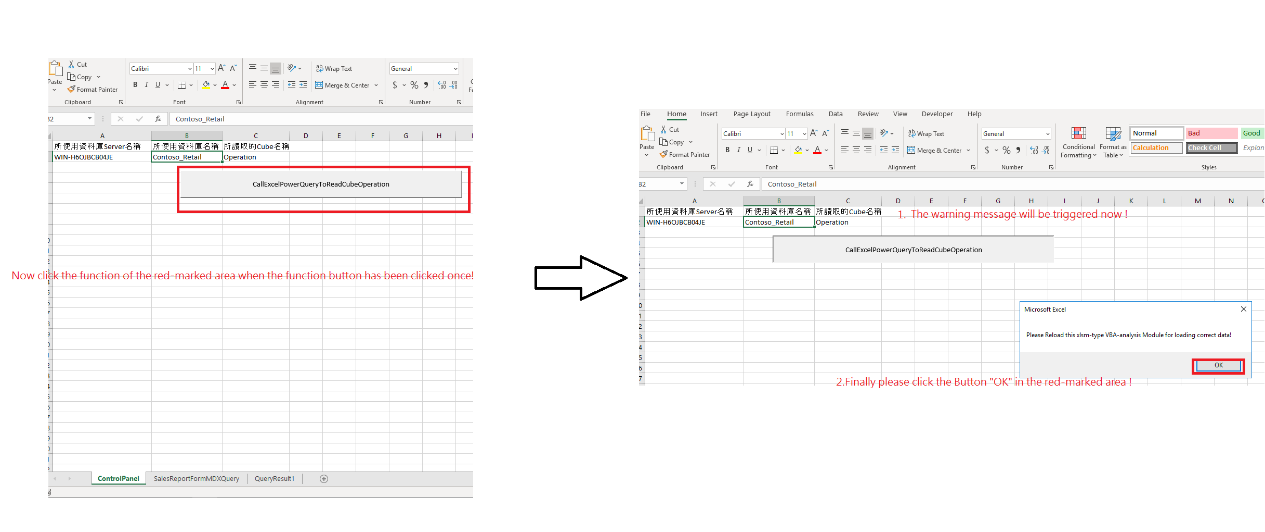


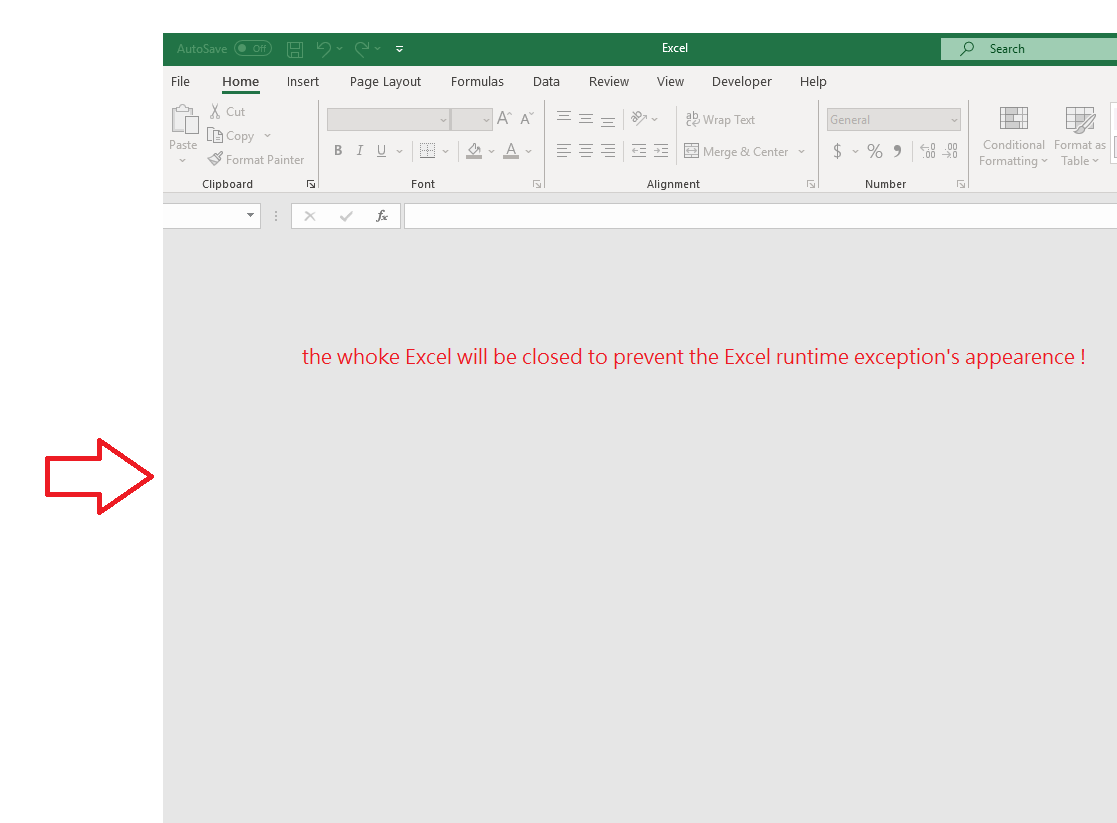
圖二(如下圖所示操作，將以下要分析的VBA成品程式載入Excel VBA開發環境):



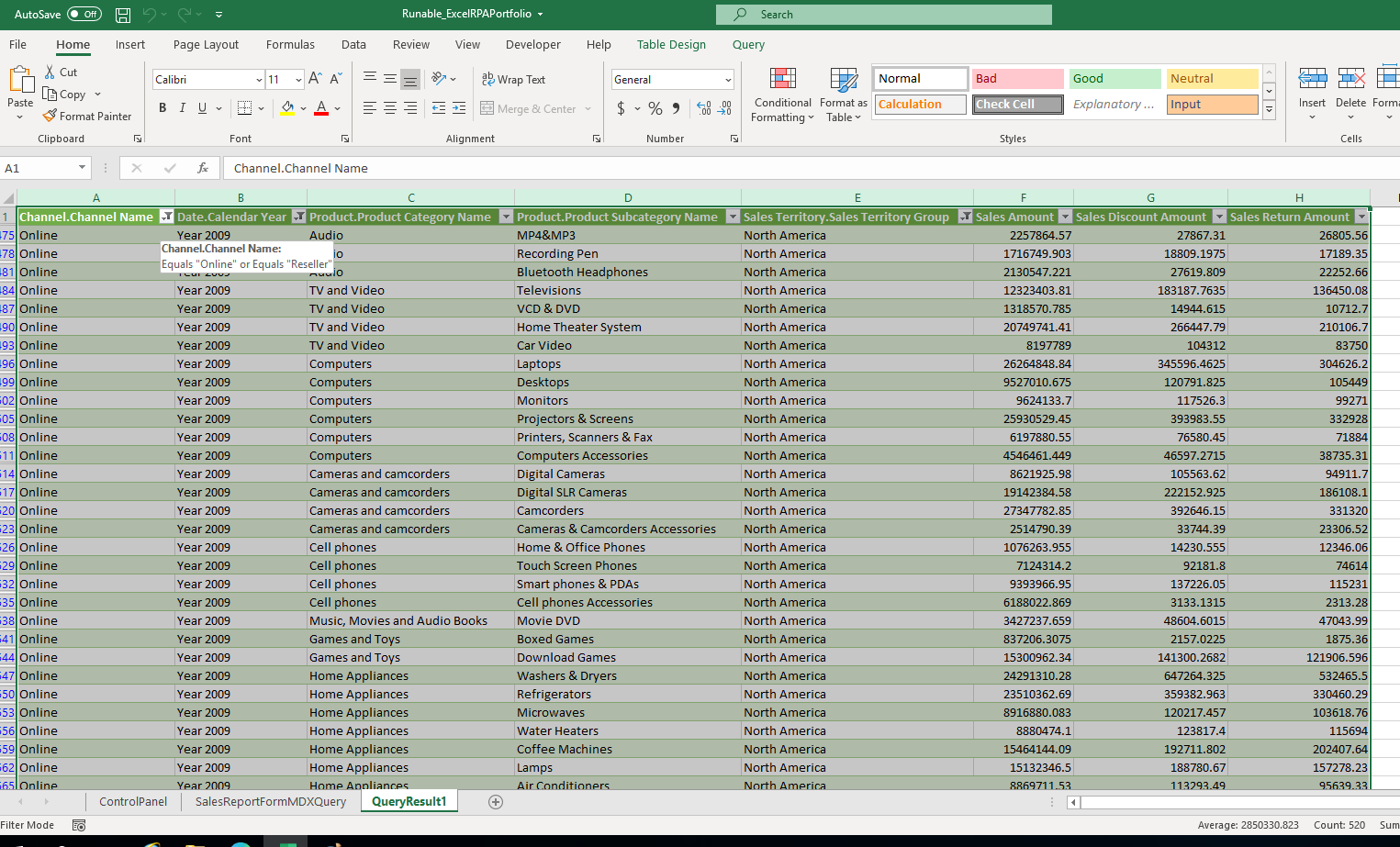


接下來就講解一下前面提到以下圖片「ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_4-5.png」，該圖片就是放在下一層以下名稱「VBA\_AutomationPortfolio\_AboutPowerQuery」的獨立目錄下→至於該程序區段相關VBA開發專業資訊，詳情請見該圖片檔本身的文字說明→那麼一旦「ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_4-5.png」的程式碼被觸發的話，就會出現如下圖所示的結果畫面:

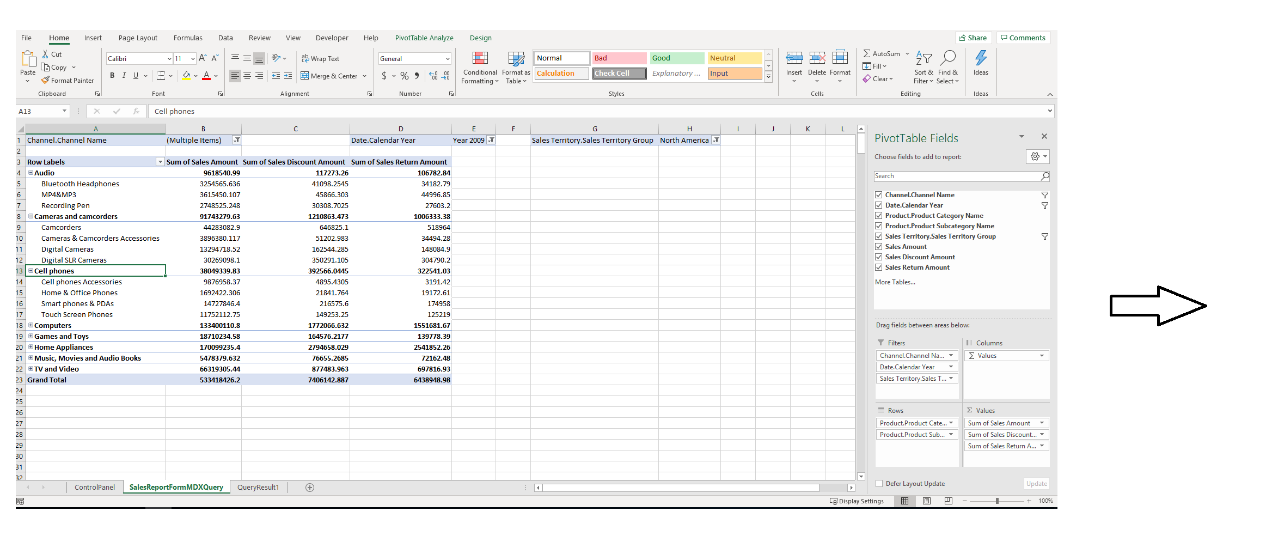


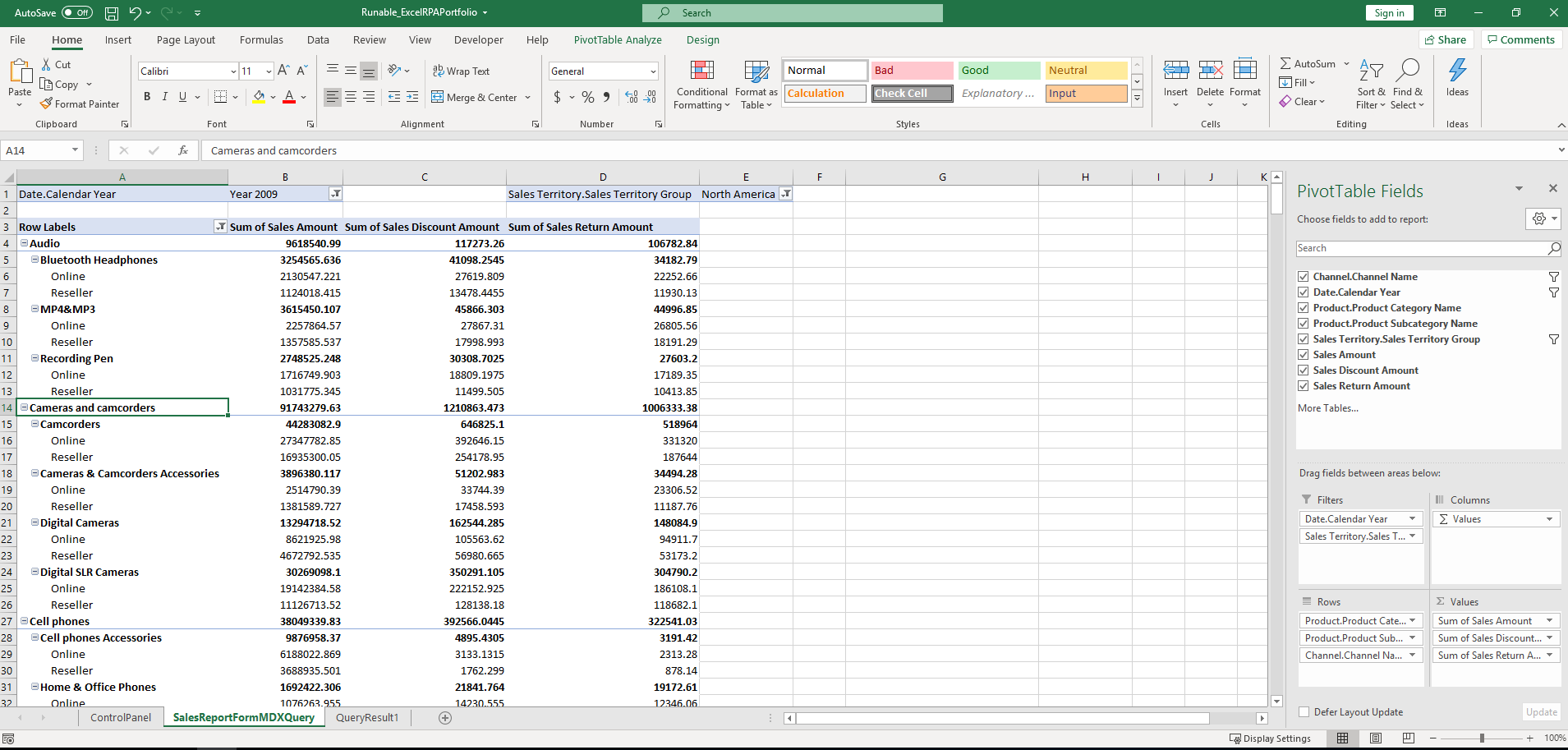


1. 圖片檔案『ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_6-8.png』Power Query自動化資料讀取程式區塊→其相關實作細節，詳情見該圖片說明文字→因此總而言之，目前實作完畢的成品，就是具備如下圖所示(流暢整合十多個條件過濾操作去產生成品的功能)



1. 圖片檔案『ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_9-14.png』Pivot Table自動化資料呈現可以互動操作程式區塊→其相關實作細節，詳情見該圖片說明文字→因此總而言之，目前實作完畢的成品，就是具備如下圖所示一系列互動展現功能:





預知詳情可以參考「VBA\_RPA-PowerQueryModule\_PivotTableDynamicVerify1-3.png」至「VBA\_RPA-PowerQueryModule\_PivotTableDynamicVerify14-15.png」總共5張圖片一系列展示→這其中也充分展現「ExcelVBACodeView\_InVBA-RPA-PowerQueryModule\_9-14.png」這一張圖片193行程式的一系列功能(展現約1/3左右的條件篩選調整變化)