

出國報告（出國類別：進修）

生態資訊系統與監測技術建置研習與參加  
US LTER資訊管理員與科學家年會

服務機關：行政院農業委員會林業試驗所

姓名職稱：鄭美如 助理研究員

派赴國家：美國

出國期間：95年06月15日～95年09月26日

報告日期：95年12月25日

# 目 次

一、目的-----	2
二、過程-----	4
三、研習內容-----	6
四、心得-----	22
五、建議事項-----	23
六、附錄-----	24

附錄一：EML, XSLT 與 R 的連結應用程式

附錄二：R 簡介與入門教學投影片

附錄三：R 進階教學投影片

附錄四：KML 簡介教學投影片

附錄五：美國長期生態研究網資訊管理員年會與科學家年會報告投影片

附錄六：美國長期生態研究網資訊管理員年會與科學家年會展示海報

## 一、目的

資訊技術的發展對研究成效影響極大，尤其是對現今的生態研究。許多的生態問題因為面向複雜、資料整合困難，因此在研究上常面臨資料分離，難以分散的個別資料解釋複雜的生態現象，尤其是長期性所蒐集的生態資料，而資訊技術的應用則解決了這方面的難題。新科學觀的生態研究主張跨領域、集體的資料整合以及不同空間與時間的資料再利用，並利用這些龐大的資料重新解釋許多生態現象與問題。這樣的研究觀已非以往使用的工具可以完成，例如偏遠山區的調查，已可透過 **Gird** 技術以無線傳輸的方式，將資料幾近即時傳回實驗室；許多複雜的生態系動態模式，可聯結數以萬計的個人電腦共同運算。但這些工作都必須基植於完整的資料庫系統與管理，因此生態學勢必面臨學習與開發運用新資訊技術的抉擇，且這部份工作不可能由資訊人員完全代勞。

美國長期生態研究網的科學家們已實現了這個觀念，並且開創出生態研究的新領域。格網運算(**Grid Computing**)是新興的資訊技術，藉由高速網路整合電腦系統，以分散式資料分析處理功能使虛擬組織的成員得以進行大規模學術合作，美國長期生態研究網，透過與資訊人員合作，建立了資訊管理員與資料庫系統合作研究兩制度，將生態研究的尺度與領域擴大。此外，再經生態學家與資訊人員合力開發 **Ecological Metadata Language (EML)**，使龐雜的生態資料有共通的標準與語言，整合成以網站服務之資料共享方式，讓生態研究網以往之資料處理、分享與整合的困難逐漸獲得改善，使得資料之再利用及創造更珍貴的價值得以實現。所以運用格網技術可解決長期生態研究所面臨的資訊技術瓶頸，在新一代網際網路技術服務純熟之後，生態研究已邁向一新的里程。

台灣目前生態研究學界之各項研究資料均為研究者各自所有，且小尺度，亦無統一的格式，無法相互分享與做進一步的整合分析，進而創造與激發大尺度的研究，實在是浪費收集的龐大資訊。台灣若要與世界接軌，資訊的即時化與整合為必行之工作，讓資料分享有一個運作良好的平台是為迫切之事。生態資訊服務系統的建立，可以整合台灣長期生態研究的各種資料，讓資料分享有運作良好的平台，提昇研究效率，不僅服務生態學家，更可以提供生態資訊給教育界及一般民眾有用可靠的生態資訊，擴展

生態研究價值的層面，最為重要的是與國際的交流合作與共同研究將更為頻繁。

目前林試所正積極發展生態資料庫之整合與網站服務之建構，以擴大生態研究之範疇與深度，並期營造資料共享之優質研究環境，林試所目前雖已培訓初步之生態資訊網站服務架設與管理及 EML 工具使用之人才，但仍需進一步研究與學習其細節，如 EML 之撰寫、資料分析工具與 EML 之結合、無線網路傳輸與資料轉換整合、地理資訊系統整合生態資料、資料分享機制建立等等。

本計劃之研究目的即為培訓國內優秀之年輕生態資訊管理人才，使台灣在此一領域與美國同步，持續領先亞洲暨太平洋地區其他各國。另外，藉由習得之技術，逐步建置完善之生態資訊管理平台，使研究資訊可以即時取得並有運作良好的分享機制，使研究資訊之生命得以延續，進而提昇研究之廣度與深度。

## 二、過程

本計畫為本所與美國國家科學基金會、美國長期生態研究網之合作計畫，美方已補助我方派遣人員在美之生活費用，派遣人員以具生態學背景人員，學習進階生態資訊系統建置與管理實務各三個月，以其較為成熟之經驗，配合理論，經由實際的參與計畫及實做中學習相關資訊技術，以期成為生態學者與資訊工作者的界面角色，並能熟悉更深入之生態資訊管理技能，使生態資訊分享服務系統運作的更完善。

研習地點為美國長期生態研究網維吉尼亞海岸保留地（Virginia Coast Reserve Long-Term Ecological Research, VCRLTER）之試驗站，研習內容與工作時程如下：

1. 3月~6月派員學習野外無線傳輸即時資料的轉換儲存與網站服務應用（無線偵測網），藉由無線資訊傳輸技術，使資料的收集更為即時有效，不僅節省時間與人力成本，透過資料的管理與控制，提供本地即時研究資訊與國際研究資訊交換的機制平台。
2. 6月~9月派員學培養撰寫與製作 EML 文件的能力，使之對 EML 之架構與組成有更深入之了解，重點在於 EML 文件與各種分析工具之應用研究，提升生態資訊分享之價值。參加美國長期生態研究網科學家年會與資訊管理員年會，吸收獲取廣泛與最新的生態資訊管理技術。
3. 10月~12月派員研習進階地理資訊系統(GIS)，除使用 openGIS 建立進階地圖網站服務外，並研習 GIS 交換標準與 GIS 平台程式撰寫能力，進而結合生態資料庫，擴大資料庫應用範圍。

本報告為第二位派遣人員(鄭美如)6月~9月期間之研習報告，其研習時程如下：

日期	起訖地點	工作任務
6/15(四)	台北→舊金山→亞特蘭大	去程轉機
6/16(五)	亞特蘭大→夏洛特維爾	去程
6/17~9/17	夏洛特維爾 (Charlottesville)	研習期間
9/18(一)	夏洛特維爾→亞特蘭大→丹佛	前往研討會地點
9/18~9/23	丹佛/艾絲特斯公園(Estes Park)	參加會議
9/24(日)	丹佛→鹽湖城→舊金山	返程轉機
9/25(一)	舊金山→台北	返程
9/26(二)	抵達台北	返程

### 三、研習內容

美國長期生態研究網維吉尼亞海岸保留試驗地（Virginia Coast Reserve Long-Term Ecological Research, VCR LTER）之資訊管理系統為美國 26 個長期生態研究試驗地相當傑出的，本所生態資訊研究小組自 2004 年起即向該站之資訊管理員 Dr. John H. Porter 學習生態資訊管理之相關技術，進而發展出長期技術交流與合作之關係。此次的出國研習計劃是針對目前台灣組所需要加強之生態資訊管理技能做學習內容之規劃，著重在生態後設資料語言之編寫與其應用開發技能之訓練，另外還包含管理系統運作之自動化工作排程編寫，並共同規劃 2007 年將在韓國舉行之亞洲暨太平洋地區長期生態研究網第三次生態資訊管理研討會之相關事宜。在 VCR LTER 研習結束後轉往科羅拉多州 Estes Park 參加美國長期生態研究網科學家年會與資訊管理員年會，廣泛的獲取美國長期生態研究網資訊管理系統與最新發展，及未來發展趨勢。以下是研習的目標、學習項目與工作內容：

#### A. 目標（Goals）

1. 利用 Linux 伺服器建置平台，服務研究社群
2. EML 應用
  - a) 可以自 EML 文件檔案裡抽取其中的資訊並應用
  - b) 發展 EML 線上分析與應用工具

#### B. 學習內容（Need to learn）

1. EML 細節與使用除 Morpho 以外的編輯工具編輯 EML
2. XSLT
3. KML
4. R 程式撰寫

#### C. 規劃工作

1. 規劃 2007 年在韓國舉辦之 EAP-ILTER 資訊管理研習會
2. 規劃後續派遣成員接續研習之事宜

#### D. 參加美國長期生態研究網資訊管理員年會與科學家年會

## (一) 生態後設資料語言(EML)的層級內容

EML 是用標籤(tag)來標示資料的每個細節，如資料標題使用<title>標籤將標題資料切出，在標題前後加上<title>與</title>標籤，斜線表示該標籤結束。所以一份 EML 文件包含了描述文件內容的各種標籤，由每段標籤大約可以知道其中所描述的文字內容涵意。

美國長期生態研究網的 EML 工作小組依據 EML 文件中所包含的內容資訊，將 EML 文件分成六個層級，每一層除了前一層的資訊外，再加入更進一步的資料，為累加的形式。第一層為 Identification(鑑別)，僅包含資料的最基本內容，如標題、所有者、聯絡人、關鍵字與摘要等；第二層為 Discovery(發現)，在第一層的基本資料裡再加入該研究的時、空與分類等資訊；第三層為 Evaluation(評估)，為加上資料使用權、計畫細項、研究方法與資料表的描述等；第四層為 Access(存取)，在第三層的內容上加資料存取之細節與實體；第五層為 Integration(整合)，增加了完整的資料屬性與內容，並具備資料品質的控管；最後第六層為 Semantic Use(語意用途)，為增加了資料語意上的說明，讓資料使用更正確，但目前仍在發展階段。以下分別詳述前五個層級的內容：

### 1. 鑑別層級

鑑別層級包含最基本的資料集(dataset)訊息，包含資料集的識別碼(ID)、標題、所有者、後設資料提供者、其他合作的研究者或組織、資料公開與否、摘要、關鍵字、資料描述的網址、聯絡人(該份資料集的資訊管理者)、出版者等資訊。在 EML 文件裡，這些元素分別以下列的標籤表示，說明如後：

packageId代表該文件的ID，且該ID為唯一編號。在美國LTER還有另一標籤<alternateIdentifier>代表ID，其中顯示的編號可以看出是哪個試驗站的資料集。

<title>資料集的標題，標題可以為計畫名稱，也可以將該資料集發生的時間地點等資訊標示於此。

<creator>資料集的所有者，這部份包含許多相關標籤以記錄所有者的資訊，<individualName>所有者姓名，這項下包含<surname>姓、<givenName>名、<salutation>稱謂等，<organizationName>所屬組織



名稱、<address>住址、<phone>電話、<electronicMailAddress>電子郵件信箱、<onlineURL>網頁連結。

<metadataProvider>後設資料提供者，在此項下包含<organizationName>組織名稱、<address>住址、<phone>電話、<electronicMailAddress>電子郵件信箱、<onlineURL>網頁連結。

<associateParty>合作者，列出所有合作者及其相關資訊，其所包含的標籤如前兩項。

<pubDate>資料釋出日期，為該份資料集資料線上釋出的日期宣告。

<abstract>摘要，此一部份內容是文字描述，在 Metacat 資料庫(一種 XML 的資料庫)裡是可以被全文搜尋的，所以可以在這一節裡簡單描述研究分類資訊、研究方法、或表列變數等。

<keywordSet>關鍵字集，此一部份若試驗站要建立關鍵字集讓該試驗站人員選用，則規定包含需含有試驗站縮寫、研究領域的編碼、地點、研究網縮寫、組織名稱與經費來源等資訊。若為一般研究人員則只需自行建立關鍵字即可。美國 LTER 已有關鍵字類別，如分類、尺度、組織層級、評估、生態與棲息地等鍵字集供研究人員選用。

<distribution>散佈，定義該資料集的資料是否可以存取，其中包含<online>線上、<offline>離線及<inline>只附加在 EML 裡三種。若為在線上可存取，則還要包含<url>標籤，宣告資料所在。

<contact>聯絡人，其細項跟資料所有者一樣。

<publisher>出版者，若為美國 LTER 研究站則需要包含所有的細項，包括<organizationName>、<address>、<phone>、<electronicMailAddress>及<onlineUrl>等。

## 2. 搜尋層級

此一層級的後設資料必須儘可能包含資料集的其他資訊，如時間、分類、地理位置等，以<coverage>標籤為基礎，其下包含<temporalCoverage>(時間範圍)、<taxonomicCoverage>分類資訊與<geographicCoverage>地點範圍等子標籤。

<coverage>範圍，用來標示資料集裡資料產生的時間、地點與分類等涵蓋的範圍。這部份提供對於該資料集更精確的查詢資訊。

<geographicCoverage>地點範圍，用以標示該資料集所涵蓋的地理範圍，其中包含<boundingCoordinates>經緯度邊界、<boundingAltitudes>海拔高度範圍與<geographicDescription>地理描述等。另外還有<datasetGPolygon>用來定義經緯度邊界無法說清楚的地方，如大範圍內有其他地區並不包含在該研究範圍內的情況，而<altitudeUnits>則用以標示高度單位。

<temporalCoverage>時間範圍，基本上以”日”為單位，可以是單一時間點<singleDate>，可以是一段時間<rangeOfDates>，若為持續不斷直至哪天結束，則可以用<endDate>標籤表示。

<taxonomicCoverage>分類資訊，在此一項下描述的是資料集所研究的對象的分類資訊，<taxonomixClassification>用來標示分類階層，在動植物學裡的界門綱目科屬種階層。

<maintenance>維護，用文字敘述來描述資料表的更新頻率與資料調查頻率。

### 3. 進階使用層級

在此一層級應包含計畫、研究方法、協定及智慧財產權等細節，讓潛在使用者可以依據這些訊息遵循這些法則來使用資料集與資料以進行他們的研究或做整合分析，這一部分亦須包括所有資料表內資料欄位基本的描述。

<intellectualRight>智慧財產權，定義包含研究者所屬的研究站的資料存取協定與個別資料集特殊協定。

<methods>研究方法，此一部分用來清楚的描述資料集重頭到尾所有的研究方法的描述，甚至包括研究事前的準備工作與程序或過去的直接相關研究程序描述。

<methodStep>研究方法步驟，<proceduralStep>、<description>、<para>等標籤用來定義研究方法的文字描述，而<citation>引用與<protocol>協議為延伸的說明，在<protocol>之下要具備的內容以

<title>標題、<creator>創造者、<pubDate>公開日期、<abstract>摘要、<keywordSet>關鍵字集及<distribution>等標籤來定義其中所含的協議，且在<distribution>之下用<online>及<url>來說明該協議文件所在處。

<methodStep><instrumentation>實驗步驟裡的儀器資訊，其中的內容以文字描述，包含儀器的基本資料與量測方法與單位與量測值誤差範圍。

<methods><sampling>取樣，標示研究方法裡所用的取樣方法及其相關資訊。

<studyExtent>延伸研究，資料集的產生若有其他延伸研究支持或特別的資訊與其資料集相關，則用此一標籤標示。

<samplingDescription>取樣描述。

<spatialSamplingUnits>空間取樣單元，這一部分是選擇標籤，用於有特殊空間取樣時，即有好幾個取樣區塊時用之。

<project>計畫，為進一步關於該計畫的一般資訊，如果該計畫與其他計畫相關，則還須有<relatedProject>相關計畫標籤。在美國 LTER 研究站裡，此一部份最少要涵蓋<title>、<personnel>、<abstract>、<distribution>、<fundingSource>等資訊。

<studyAreaDiscription>研究區域的描述，用以標示沒有涵蓋到的研究區域資訊，如氣候、地理環境等資訊。

在此一層級另一重要的部份為資料表的描述與各項欄位的定義，其包含的標籤說明如下：

在 dataTable/entityGroup 項下，為表格基本資訊：

<AlternateIdentifier>額外識別碼，為選擇性標籤，如果資料集亦存於其他資料管理系統，則可能有其他系統給的編號。

<entityName>名稱，為表格名稱或原始 ASCII 檔案名稱。

<entityDescription>描述，通常是更詳細的描述標題，但不一定要跟資料集標題相關。

在 dataTable/attributeList 項下，即是表格裡各項欄位的定義：

<attributeName>屬性名稱，為資料表裡各欄名稱，通常會用簡單的縮寫代表，如 DBH 表示該欄資料為“胸高直徑”。

<attributeLabel>屬性標記，替屬性名稱做簡要說明。

<attributeDefinition>屬性定義，清楚的描述該屬性所代表的意義。

<storageType>儲存型態，一般而言為小數點、整數或字串。

<measurementScale>量測尺度，定義該欄位採用哪種量測尺度，是名義尺度(nominal)，序列尺度(ordinal)，等距尺度(interval)、比例尺度(ratio)還是日期時間(date-time)。該五種標籤所定義的內涵說明如下：

<nominal>若為數字，可能表示分類的代碼，這些不同的值是沒有順序上的區別；或者直接用一般文字表示，如男、女。

<ordinal>序列尺度的值具有順序上的意義，所有順序尺度的值亦為類別，但順序尺度間的值具有一個特別的關係，例如，低、中、高等，這些序列尺度彼此之間沒有距離上的差別(只問二數量間的相對大小，不問其差距)。

<interval>其資料間具有相同的測量單位，因此資料間在尺度上是可以比較的。區間尺度的資料可以隨意設置起點(並不一定是從零開始)，所以並沒有“零”的測量觀點亦無比例上的意義，其數字加減有意義，乘除無意義。

<ratio>比例尺度是一個具有零點的區間尺度，起始點在零，此零點表示被量測值為 0，因此比例值具有意義。

<date-time>日期時間亦希望符合 ISO8601 標準，如 YYYY-MM-DD，或者 YYYY-MM-DDThh:mm:ssTZD。

<missingValueCode>缺失值代碼，為選項，但如果選用，則必須將所有的缺失值代碼都列上。

#### 4. 存取層級

若要 EML 文件到此一層級，則須包含所有的必須資訊及可下載表格資料，即使表格裡的屬性並沒有定義得非常完善。在此可以只讓資料所有

者指定可以下載的群體或個人。此一層級特別需要定義清楚的有以下幾個標籤：

<access>存取，在 Metacat 系統裡，可以指明<allow>允許哪些使用者的 ID 及組織。

在 entityGroup/physical 項目裡須做進一步說明的標籤有：

<objectName>目標檔名

<externallyDefinedFormat>外部定義格式，如軟體、製造商、版本等資訊，如 Microsoft Excel 2002。

<distribution>傳播，宣告可連結下載的 URL，有可能只是內部網路。

## 5. 整合層級

此一層級的後設資料應支援資料的電腦存取與程序，並包含資料集最完整的描述。在此增加的標籤在 methods/qualityControl 與 dataTable/constraint 項下。在 dataTable/attributeList 底下必須有完整的描述，包括所有的選項，如<precision>資料精確度，即資料的有效小數位數，而<constraint>限制是描述該表格在整合時應注意的事項與限制。

### (二) EML 的編輯軟體：oXygen XML editor

oXygen XML Editor 是一種操作簡單並且功能強大的 XML 察看與編輯之商用軟體，是一種基於 Java 的 XML 編輯器，支援 XML, XSL, TXT, XSD, DTD 文件，能自行校驗 XML, XSL, XSD 代碼，提示腳本文件的錯誤，且 Oxygen 能自動完成 XML 結構的結束標籤，由於它提供了對 XML 編輯特性的完整覆蓋，因此該軟體在編輯 XML 文件的使用上非常普及。oXygen 不僅包括基本的 XML 編輯，它還包括 XML Schema 編輯器、XSL/XSLT 編輯器以及 XSL/XSLT 重構功能。同時，它還提供了 XSL, XSLT 和 XQuery 調試功能。此外，它還能與 Xerces、MSXML.NET 和 Saxon SA 等大型驗證引擎一起工作，並能處理 SGML DTDs、RELAX NG 以及 XML Schema，同時支持批量驗證以及文件之間的轉換。

oXygen 可以簡單的在單一文件的多種視圖間切換，而其間會與所編輯文件同步自動維護一個樹狀外形。其他視窗包括一個 XML 原始編輯器、一

個 XSLT 調試器、一個 XQuery 調試器以及一個樹形視窗編輯器。該樹形視窗編輯器代表了一個層次的樹形式的文件，而不是正在執行的文本。

oXygen 也可以執行 XML Diff 和 Merge 來比較或聯合同一文件的多個版本，可以應用拼寫檢查功能，與 FOPs (規格化物件處理器)一起工作，並且跨平台，可以安裝於 Windows、Mac 或 Linux 平台上。

因為 EML 文件本身便是一種 XML 文件，符合 XML 規範，所以 oXygen 是一種操作簡單且學習容易的自行編輯與修改 EML 文件的工具。

### (三) 擴展樣式表轉換語言(Extensible Stylesheet Language Transformations, XSLT)

XSLT是擴展樣式表轉換語言(Extensible Stylesheet Language Transformations)的簡稱，XSLT是XML家族技術一員，同時也是XML應用語言，目的是在轉換XML的結構，改變原本的XML文件的結構組織，亦即將原文件的所有數據文字或者部分數據文字生成另外的XML文件或者其他可直接顯示或列印的文件格式，如 HTML文件, RTF文件或者TEXT文件。在轉換過程中涉及一些HTML文件中的固定標籤、移動文本內容及對文本內容的排序與挑選等。被轉換的原XML文件具有樹狀結構，而XSLT語言是聲明性的語言，即XSLT程序本身只是包含了一些轉換規則的文件，而這些規則可以被遞歸地應用到轉換過程中而產生新的文件。

學習 XSLT 的目的在於開發 EML 文件的後續應用，透過 XSLT 的轉換，可以將取得之 EML 文件，從中抽取有用的資訊，形成另一個文件檔案，使之可以應用在其他軟體上。就生態研究而言，資料的統計分析是為十分重要的一環，我們可以透過 XSLT 文件的編寫，將 EML 文件中相關的資訊抽取出來，再添加所應用之統計軟體的程式語法，形成該統計軟體的程式檔案，便可以在該統計軟體裡直接將該程式檔案匯入，加進原始資料便可以做統計分析。這樣的好處是，一份 XSLT 文件便可以適用於各種不同的 EML 文件，以達到自動產生統計分析程式之境界，在生態資訊管理上的便利與應用效益是一大進步。

#### (四) R 統計軟體

R是一個有著統計分析功能及強大作圖功能的軟體系統，是由Ross Ihaka和Robert Gentleman<sup>1</sup>共同創立。R語言可以看作是由AT&T貝爾實驗室所創的S（為一種商業統計軟體）語言發展出的一種方言，因此，R即是一種軟體也可以說是一種語言。

R是在GNU協議General Public Licence下免費發行的，它的開發及維護現在則由R開發核心小組R Development Core Team負責。R為跨平台，可以安裝於Windows, Linux及Macintosh上。

R內含了許多實用的統計分析及作圖函數。作圖函數能將產生的圖片展示在一個獨立的視窗中，並能將之保存為各種形式的檔(jpg, png, bmp, ps, pdf, emf, pictex, 其圖檔形式取決於作業系統)。統計分析的結果也能被直接顯示出來，一些中間結果(如 $P$ -值，回歸係數，殘差等)既可保存到專門的檔中，也可以直接用作進一步的分析。

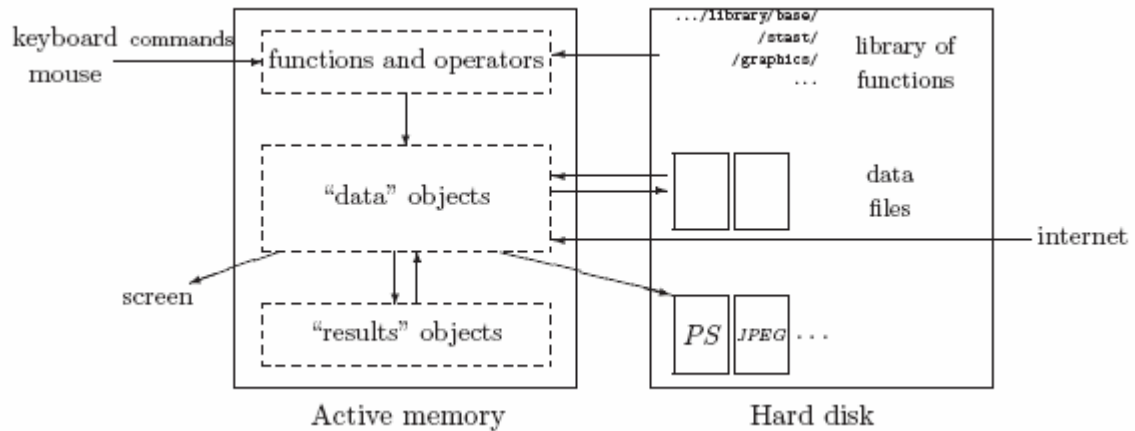
在R語言中，使用者可以使用迴圈語句來連續分析多個資料集，也可將多個不同的統計函數結合在一個語句中執行更複雜的分析。R使用者還可以借鑒網上提供的用S編寫的大量程式，而且大多數都能被R直接調用。其實，R的一個非常突出的優點正是它的靈活性。一般的軟體往往會直接展示分析的結果，而R則將這些結果都存在一個物件(object)裏面，所以常常在分析執行結束後並不顯示任何結果。使用者可能會對此感到困惑，其實這樣的特點是非常有用的，因為我們可以選擇的從結果中只抽出我們感興趣的部分。例如，我們要運算20個回歸分析而只想比較其回歸係數，在R中就可以選擇只顯示所有分析得出的回歸係數，這樣結果僅僅占了一排，而用有些軟體可能會一下打開20個視窗。而在下面的章節中，我們會看到更多能展示R相比傳統軟體更為靈活優越的例子。

相對其他同類軟體，它的特色在於有效的資料處理和保存機制、完整的陣列和矩陣運算操作符、連貫而又完整的資料分析工具，其圖形工具可以對資料直接進行分析和展示，同時可用於多種圖形設備，它是一種相當完善，簡潔而又高效的程式設計語言。

當R運算時，所有變數，資料，函數及結果都以物件(objects)的形式存在電腦的活動記憶體中，並冠有相應的名字代號。我們可以通過用一些運算

符(如算術，邏輯，比較等)和一些函數(其本身也是物件)來對這些物件進行操作。

在R中進行的所有操作都是針對存儲在活動記憶體中的物件的, 因此就不涉及到任何暫存檔案夾的使用(下圖)。對資料，結果或圖表的輸入與輸出都是通過在對電腦硬碟中的檔讀寫而實現。使用者透過輸入一些命令函數，分析得出的結果可以被直接顯示在螢幕上，也可以被存入某個物件或被寫入硬碟(如圖片物件)。因為產生的結果本身就是一種物件，所以它們也能被視為資料並能像一般資料那樣被處理分析。資料檔案即可從本機電腦讀取，也可通過網路傳輸從遠端伺服器端獲得。



R工作原理示意圖

所有能使用的R函數都被包含在一個函式庫(library) 中, 該函式庫存放在本機電腦的R HOME/library 目錄下(R HOME 是最初安裝R的地址)。這個目錄下含有具有各種功能的套件(packages )，各個套件也是按照目錄的方式組織起來的。其中名為base的套件可以算是R的核心，因為它內嵌了R語言中所有像資料讀寫與操作這些最基本的函數。在上述目錄中的每個套件內，都有一個子目錄R，這個目錄裏又都含有一個與此套件同名的檔(例如在base套件中，有這樣一個文件R HOME/library/base/R/base)。該檔正是存放所有函數的地方。

關於詳細的R使用介紹，請參考附錄二、三。



## (五) Keyhole Markup Language(KML)

Keyhole Markup Language 亦為 XML 家族成員，有固定的語法與結構。KML 也是由各式各樣的標籤名稱來定義文件所有的內容，就像大家熟悉的網頁語言 HTML(HyperText Markup Language)一樣，用標籤來讓機器知道該在網頁上顯示什麼文字、線條或顏色。KML 文件跟其他 XML 家族最大的不同處是他以地理資訊的描述為主，如點、面、圖形與立體空間的定義與描述。而該文件亦承襲 XML 家族的特性，是人跟機器都可以讀的懂的。KML 文件內容可以在 Google Earth 的地圖上完整呈現，也就是說 Google Earth 可以處理所有的 KML 文件，然後在 Google Earth 的視窗上呈現出來，換句話說，Google Earth 就像 KML 文件的瀏覽器。另一方面，在 Google Earth 裡所有的地理樣貌都可以被描述進 KML 文件。KML 除了可以準確的標的座標位置外，亦可以設定隨意的高度與角度來觀看該地點，並加入圖案或文字、畫上 3D 立體圖、可以設定網頁連結等。

KML 與 EML 文件的連結在於 EML 文件裡存在研究的地理空間位置，透過其他程式的轉換與應用，可以抽取 EML 文件中的地理空間位置，轉換成 KML 文件而展示於 Google Earth 的視窗裡，也就是說，資訊管理系統不必取得大量的衛星影像與圖層，透過免費使用的 Google Map 與使用者端安裝的 Google Earth，便可以清楚的展示研究的地理資訊，節省了大量的圖層與衛星影像取得的成本與時間外，更增加了管理者與使用者的便利性。

關於 KML 的介紹，請參考附錄四。

## (六) 美國長期生態研究網資訊管理員年會與科學家年會

美國長期生態研究網資訊管理員年會是每年舉行一次，而科學家年會是每三年舉行一次，故今年地兩個會意在美國科羅拉多州的 Estes Park 的 YMCA 一起舉行，9/18~9/20 先行舉行管理員會議，全美 26 個試驗地的資訊管理員及其他來自十個國際長期生態研究網(包括墨西哥、巴拿馬、委內瑞拉、南非、澳洲、馬來西亞、日本、以色列、奧地利與台灣)的十數位資訊管理員，共五十餘人與會。會中除邀請國際長期生態研究網的資訊管理員報告分享其資訊管理現況外，對於美國生態資訊管理系統現況報告與分

析及未來發展方向均有詳細的討論。其中以「**Cyberinfrastructure (CI)**」議題著力最多，以下即針對**Cyberinfrastructure**做一簡單介紹。

**Cyberinfrastructure** 是提供一個寬廣的整合性技術以支援逐漸複雜、大尺度與合作性的科學性議題，即是透過一個廣闊的電腦網路資源架構與系統，讓資源可以相互流通，相互利用，讓想要的資訊與資料隨手可得。**Cyberinfrastructure** 已經成爲美國長期生態研究網近年來在生態資訊管理上最重要的研究發展重點，並爲此投入大量的人力與經費，要讓這系統”起動”，預估花費就超過 1600 萬美元。

CI 可以算是提供一種新的研究環境，它透過網路提供了進階資料的取得、儲存、管理、整合、挖掘與其他有關的計算與資訊處理程序服務，在科學性的應用上，CI 是一種技術性的問題解決方案，運用已授權使用的新的科學理論與知識，可以有效的連結資料、計算機與人而獲得解答。CI 整個架構包含人跟組織對系統的操作與維護、軟體的發展與支援、建置各項標準與最佳使用文件、提供其他關鍵服務如安全性與使用者支援等。

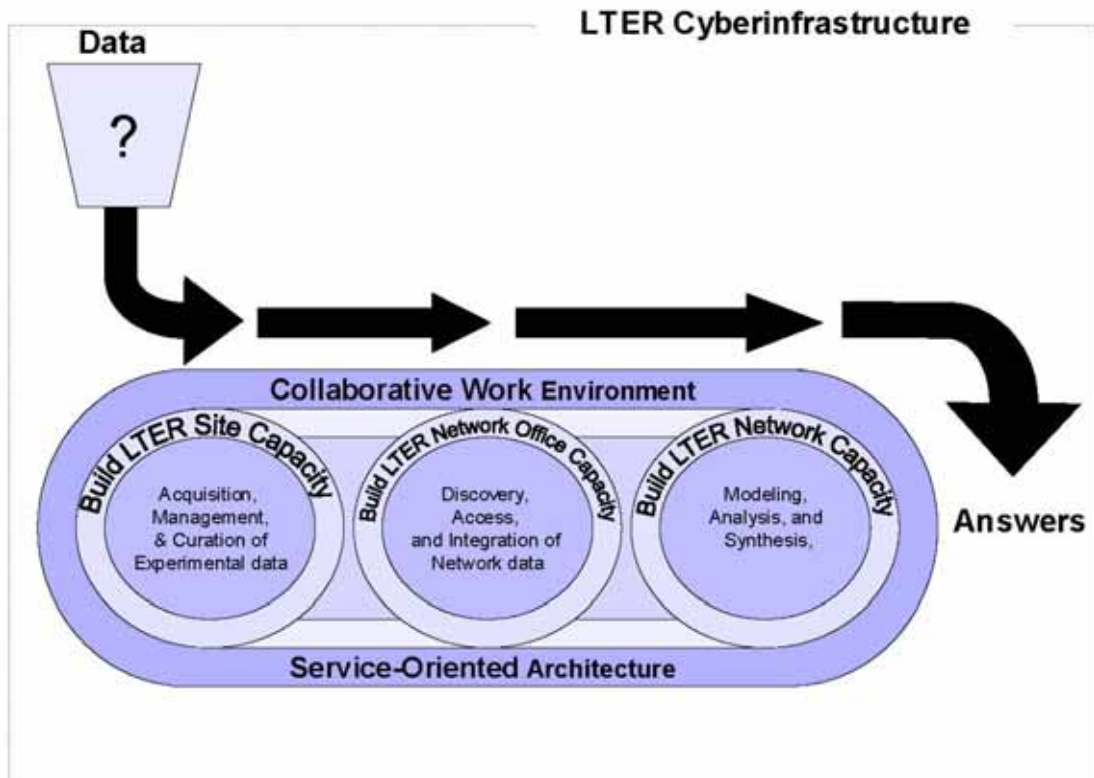
因爲 CI 所可以提供的功能服務，對於長期生態研究將有重大的突破，因爲這將是一種新的研究方法，透過 CI，研究人員可以藉由最大的資料流、綜合性資訊與所有的知識，在複合的時間與空間尺度上，促進與提升進一步的合作與綜合性的社會生態科學研究。但要達到此一目標，基礎的資訊與分享是不可或缺的，所以人與技術是最重要的兩個部份：

人—對於資訊管理員而言，資料的管理必須達到讓資料達到可以進行整合的標準，並發展一個可以綜合資料的應用與服務平台。再者，進行教育訓練，讓研究人員體認 CI 的好處，願意改善研究資訊與資料品質以達標準，並釋放出來，進而可以使用新的技術進行跨領域的資料的綜合與應用。

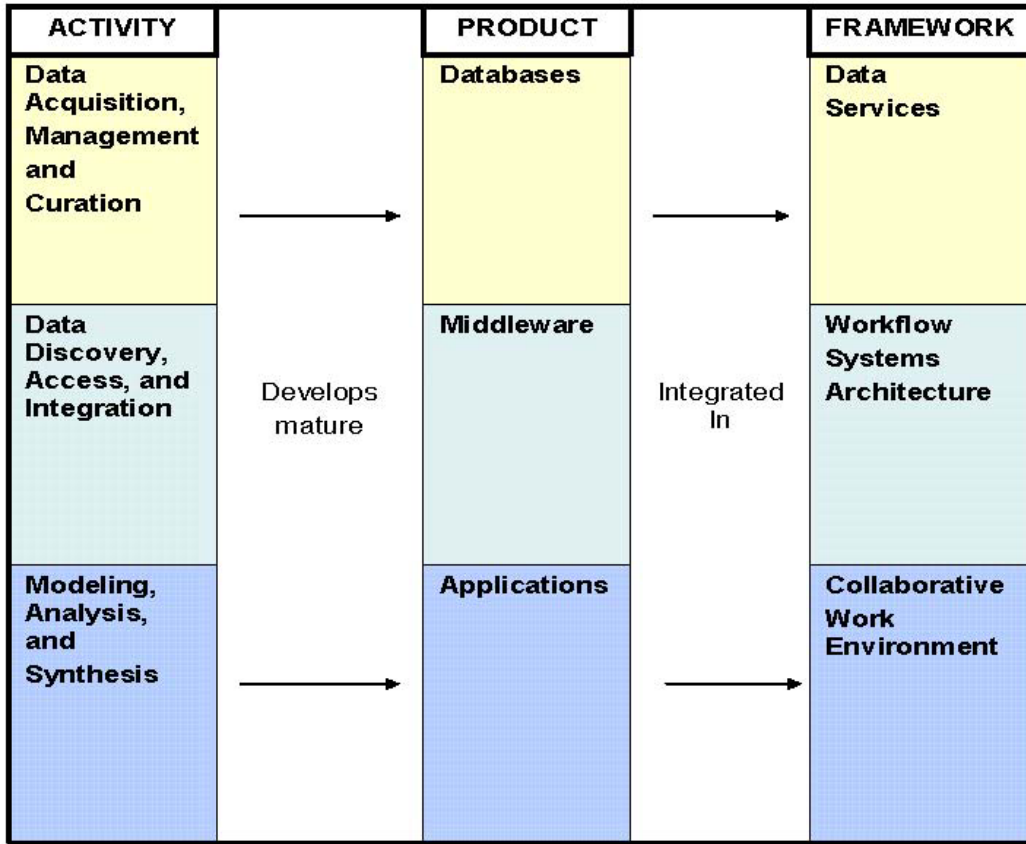
技術—一切的技術發展均朝向支援進階的合作與諮詢，讓資料的取得、一般化、產生、管理、挖掘、整合、知識呈現、分析、綜合與模型等可以輕易的獲得。

美國長期生態研究網著手建置 CI 開始，其工作項目包含讓長期資料豐富廣泛與完整、使資訊與資料標準化、制定開放資料的存取協定、涵蓋多樣的知識取得、網路辦公室的建立與進行資訊管理的研究等。這樣一個完

整的計畫可以讓資訊與知識的取得變得隨手可得，像日常生活用電一樣的便利。而研究人員的責任是在取得研究資源，獲得研究聲譽之後，研究資訊與資料的分享與回饋，充實 CI 的知識內容，讓這樣的研究環境與資訊新世代可以解答更多的疑問，發揮研究最大的效用。



將資料送進完整的合作工作環境平台，透過資源分享與一連串資料整合與分析流程，可以迅速的得到想要的答案與結果。(圖片來源：LTER Network Cyberinfrastructure Strategic Plan - DRAFT)



CI 的整個結構與建置流程，分成三個部份，每個部份都有其應有的有效資源、中介產出與最後應用的框架(圖片來源：LTER Network Cyberinfrastructure Strategic Plan - DRAFT)

此次會議中，台灣團隊除報告台灣生態資訊管理之發展現況外，亦與美國 VCR 試驗站資訊管理員 Dr. John Porter 一起分享了雙方兩年多來在生態資訊管理之國際合作模式與成果，展示林業試驗所生態資訊研究小組所建置之生態資訊管理系統。其報告投影片與參展海報詳見附錄五、六。



與 US LTER VCR 資訊管理員 Dr. John Porter 合影於 Hog Island



資訊管理員會議，透過視訊亦可以共同主持會議



科學家年會的大禮堂演講現場



於科學家年會的國際長期生態研究合作會議上分享台灣成功經驗與  
展示生態資訊管理系統

#### 四、心得

- (一) 由 **EML** 各層級之內容，不難看出各個層級對潛在使用者的使用益處，不同的層級在實際應用層面上各不相同。在鑑別與發現層級上，研究者所提供的僅僅是研究的基本資訊，使用者僅可知道有哪些研究在進行、有哪些人在研究這一領域、可能有了什麼樣的成果。但在第三層與第四層時，使用者可以進一步獲知該研究如何進行，包括收集資料的方法、資料的內容，也可進一步的瞭解原始資料的各項定義與細節。有了這些明確的資訊，研究人員可以進一步思考與他人合作的可能性，擴大研究層面，增加研究的深度與廣度。在整合層級裡，透過資料的品質控管宣告，有興趣的研究人員要與自己的資料做整合與分析變的容易且即時，讓資料可以解釋的層面更廣，卻不失其準確性。而透過智慧財產權與存取協定，雙方可以迅速獲得研究上的成果。
- (二) 美國長期生態研究試驗站對於生態資訊管理已實行一段時間，研究學者們與資訊管理員之間已經發展出一套適合自己試驗站的合作模式，不論是共同合作討論合適的系統或是由資訊管理員主導系統的開發，其研究資訊都已標準化(符合 **EML** 標準)、系統化，對於研究資料線上公開分享也都實行順暢。
- (三) 開放原始碼之應用確實已逐漸成爲世界之潮流，不斷應運而生的應用軟體，可以爲生態資訊管理系統注入新的元素，不斷改進系統的應用層面與便利使用者，在尚未有自行開發應用軟體的階段，搜尋潛在可能有用之免費軟體加以研究，用以更新與改善生態資訊管理系統是爲一可行且迅速的方式。但若在版權使用費不高，可以節省相當多的人力與時間的考量下，類似 **oXgyen** 的應用軟體亦不失爲良好選擇。
- (四) 經過持續不斷的與美國長期生態研究網 **VCR** 試驗站的交流研究與合作，台灣目前已掌握先進之生態資訊管理技術，並與美國生態資訊學界維持長期良好之互動關係，已使台灣在生態資訊管理技術上持續領先亞太各國，成爲亞太地區該領域之翹楚。

## 五、建議事項

- (一) 提升研究人員研究後設資料建置之意願：EML 除了可以實現資料分享外，更重要的是對於研究者本身有助於其資料之保存、記憶與傳承等好處。試想，一個大學或研究單位的研究室，要將過去的研究與資料清楚的讓新進人員知道，不僅繁瑣冗長且還不一定完整，若研究室有過去所有研究的 EML 文件，這件事將變的輕而易舉。美國不斷在發展一些 EML 應用軟體以擴大其應用領域，林試所亦積極改善與增加 EML 文件的進階應用的系統，此一觀念之推展可使台灣生態研究與世界先進國家的生態研究因資料使用共通的資料標準而更易於與世界交流、合作，使研究範疇國際化，創造出更多更傑出的研究成果。
- (二) 鼓勵開放原始碼之軟體的開發與應用：免費之開放原始碼應用軟體之開發雖然較為耗時，但透過分享機制，可以結合全世界的專業或業餘的頂尖程式設計師共同開發與修正，讓免費的應用軟體更臻完美，接近商業軟體所具備的功能。生態資訊管理員可以利用這些免費的應用軟體，組合出一個適合的生態資訊管理系統，服務研究人員，不僅節省經費，亦節省人力。
- (三) 持續發展生態資訊學之研究：生態資訊學研究目標是為協助研究人員可以妥善管理自己的研究資料，從資料開始蒐集前的設計到進入資料庫後與日後的分享機制建立，實現資料再利用、延續資料生命與提昇資料價值。
- (四) 加強培養台灣生態資訊管理人才：國內目前生態資訊管理員的智能雖已有一定程度，但應繼續累積建置生態資訊服務管理系統之能力，以建置更為完善的生態資訊管理系統與分享平台。



## 六、附錄

### 附錄一：EML, XSLT 與 R 的連結應用

以下是研習期間所編寫之 XSLT 文件，利用該份文件抽取所連結之 EML 文件中之研究基本資訊與表格內容屬性，產生 R 程式，所產生之 R 程式便可以直接應用於 R 統計軟體介面裡，進行統計分析，產生結果。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  This is just a first test for programing a XSLT stylesheet by Meei-ru Jeng.
  This stylesheet's goal is to try select attrubutes from an EML document and creating an R program
  that can do some general staticistic analysis.
-->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="*/dataset">
      <xsl:call-template name="DatasetDescription"/>
      <xsl:call-template name="DatasetOwner"/>
      <xsl:call-template name="AssociatedParty"/>
      <xsl:call-template name="Contact"/>
      <xsl:call-template name="DataTable"/>
    </xsl:for-each>
  </xsl:template>

  <xsl:template name="DatasetDescription">
    <xsl:for-each select="../@packageId">
      #Package ID(編號): <xsl:value-of select="../@packageId"/> Catalog System: <xsl:value-of select="../@system"/>
      <xsl:text>.</xsl:text>
    </xsl:for-each>
    #Data Set Title(標題): <xsl:value-of select="title"/><xsl:text>.</xsl:text>
  </xsl:template>

  <xsl:template name="DatasetOwner">
    #Data Set Owner(資料所有者): <xsl:text><xsl:for-each select="../dataset/creator">
      #Name(姓名): <xsl:value-of select="individualName/givenName"/><xsl:text> </xsl:text><xsl:value-of
select="individualName/surName"/>
      #Position(職稱): <xsl:value-of select="positionName"/>
      #Organization(組織名稱): <xsl:value-of select="organizationName"/>
      #Address(聯絡地址): <xsl:value-of
select="address/deliveryPoint"/><xsl:text> </xsl:text><xsl:value-of
select="address/city"/><xsl:text> </xsl:text><xsl:value-of
select="address/postalCode"/><xsl:text> </xsl:text><xsl:value-of
select="address/country"/>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

```

    #Phone Number(電話號碼): <xsl:value-of select="phone"/>
    #E-Mail(電子郵件地址): <xsl:value-of select="electronicMailAddress"/>
  </xsl:for-each>
</xsl:template>

<xsl:template name="AssociatedParty">
#Associated Party(合作者): <xsl:text/><xsl:for-each select="../dataset/associatedParty">
  <xsl:value-of select="individualName/givenName"/><xsl:text> </xsl:text><xsl:value-of
select="individualName/surName"/><xsl:text> </xsl:text> <xsl:value-of select="positionName"/> <xsl:text> </xsl:text>
  <xsl:value-of select="organizationName"/><xsl:text> </xsl:text> <xsl:value-of
select="address/deliveryPoint"/><xsl:text> </xsl:text><xsl:value-of select="address/city"/><xsl:text>
</xsl:text><xsl:value-of select="address/postalCode"/><xsl:text> </xsl:text>
  <xsl:value-of select="address/country"/><xsl:text> </xsl:text> <xsl:value-of select="phone"/> <xsl:text>
</xsl:text> <xsl:value-of select="electronicMailAddress"/> <xsl:text> - </xsl:text> <xsl:value-of select="role"/>
  </xsl:for-each>
</xsl:template>

<xsl:template name="Contact">
#Contact(聯絡人): <xsl:text/><xsl:for-each select="../dataset/contact">
  <xsl:value-of select="individualName/givenName"/><xsl:text> </xsl:text><xsl:value-of
select="individualName/surName"/><xsl:value-of select="positionName"/><xsl:text> </xsl:text> <xsl:value-of
select="organizationName"/>
  <xsl:value-of select="address/deliveryPoint"/><xsl:text> </xsl:text><xsl:value-of select="address/city"/><xsl:text>
</xsl:text><xsl:value-of select="address/postalCode"/><xsl:text> </xsl:text><xsl:value-of select="address/country"/>
  <xsl:value-of select="phone"/><xsl:text> </xsl:text> <xsl:value-of select="electronicMailAddress"/>
  </xsl:for-each>
</xsl:template>

<xsl:template name="DataTable">
  <xsl:if test="dataTable[. != '']">
    <xsl:for-each select="dataTable">

      <!--列出所有的表格欄位屬性--><xsl:text/>

#請將下面的 "請輸入資料檔實際路徑" 置換您資料所在的實際路徑(如：d:/mydata/datafile.txt).<xsl:text/>
infile<xsl:value-of select="position( )"/> &lt;- file("請輸入資料檔實際路徑", open="r")<xsl:text/>
  <xsl:choose>
    <xsl:when test="physical/dataFormat/textFormat/simpleDelimited[. != '']">
      <xsl:call-template name="readCSV"/>
    </xsl:when>
  <!--
    <xsl:when test="physical/dataFormat/textFormat/complex/textFixed[. != '']">
      <xsl:call-template name="readFixed"/>
    </xsl:when> -->
  </xsl:choose>

<!--將不同表格的欄位名稱與標籤列出-->

```

```

tmp_var &lt;- character()<xsl:text></xsl:text>
tmp_label &lt;- character()<xsl:text></xsl:text>
<xsl:for-each select="attributeList">
  <xsl:for-each select="attribute">
    <xsl:if test="attributeLabel[. != ' ']">
tmp_var &lt;- c(tmp_var, "<xsl:call-template name="attributeName"/>")<xsl:text></xsl:text>
tmp_label &lt;- c(tmp_label, "<xsl:value-of select="attributeLabel"/>- <xsl:value-of
select="measurementScale*/unit*/")<xsl:text></xsl:text>
    </xsl:if>
  </xsl:for-each>
</xsl:for-each>
labelFrame<xsl:value-of select="position()"/>&lt;- data.frame(variable=tmp_var, label=tmp_label)<xsl:text></xsl:text>
rm(tmp_var, tmp_label)<xsl:text></xsl:text>

```

<!--列出欄位名稱與標籤，如果是用代碼的話-->

```

tmp_var &lt;- character() <xsl:text></xsl:text>
tmp_code &lt;- character() <xsl:text></xsl:text>
tmp_label &lt;- character() <xsl:text></xsl:text>
<xsl:for-each select="attributeList">
  <xsl:for-each select="attribute">
    <xsl:if test="measurementScale/nominal[. != ' ']">
      <xsl:if test="measurementScale/nominal/nonNumericDomain/enumeratedDomain[. != ' ']">
        <xsl:for-each select="measurementScale/nominal/nonNumericDomain/codeDifinition">
          <xsl:if test="code[. != ' ']">
tmp_var &lt;- c(tmp_var, "<xsl:call-template name="attributeName"/>") <xsl:text> </xsl:text>
tmp_code &lt;- c(tmp_code, "<xsl:value-of select="code"/>") <xsl:text> </xsl:text>
tmp_label &lt;- c(tmp_label, "<xsl:value-of select="definition"/>")<xsl:text> </xsl:text>
          </xsl:if>
        </xsl:for-each>
      </xsl:if>
    </xsl:if>
  </xsl:for-each>
</xsl:for-each>
codeLabelFrame<xsl:value-of select="position()"/>&lt;- data.frame(variable=tmp_var, code=tmp_code,
label=tmp_label)<xsl:text></xsl:text>
rm(tmp_var, tmp_code, tmp_label)<xsl:text></xsl:text>

```

#列出變數(欄位名稱)，來自 dataTable<xsl:value-of select="position()"/> 以及這些變數的標籤 labelFrame<xsl:value-of select="position()"/> <xsl:text></xsl:text>

#代碼有被指定的變數 codeLabelFrame<xsl:value-of select="position()"/> <xsl:text></xsl:text>

attach(dataTable<xsl:value-of select="position()"/>)<xsl:text></xsl:text>

summary(dataTable<xsl:value-of select="position()"/>)

#以下是基本的分析，您可以自己加進您要的分析語法進去

<!--以下是為名義尺度(nominal)或順序尺度(ordinal)變數所做的一些基本的統計值(default statistical summaries)-->

```

    <xsl:for-each select="attributeList">
      <xsl:for-each select="attribute">
        <xsl:if test="measurementScale/nominal|measurementScale/ordinal[ != ' ' ]">
summary(as.factor(<xsl:call-template name="attributeName"/>)) <xsl:text></xsl:text>
stem(<xsl:call-template name="attributeName"/>) <xsl:text></xsl:text>
        </xsl:if>
      </xsl:for-each>
    </xsl:for-each>

<!--以下是為連續性變數所做的一些基本的統計值(default statistical summaries)-->
    <xsl:for-each select="attributeList">
      <xsl:for-each select="attribute">
        <xsl:if test="measurementScale/interval|measurementScale/ratio|measurement/datetime[ != ' ' ]">
summary(as.numeric(<xsl:call-template name="attributeName"/>)) <xsl:text></xsl:text>
        </xsl:if>
      </xsl:for-each>
    </xsl:for-each>
</xsl:for-each>
</xsl:if>
</xsl:template>

    <xsl:template name="readCSV">
      <!--將 CSV 類型之表格檔案讀進來-->
dataTable<xsl:value-of select="position( )"/> &lt;- read.csv(infile<xsl:value-of select="position()"/>, <xsl:text> </xsl:text>
    <xsl:choose>
      <xsl:when test="physical/dataFormat/textFormat/simpleDelimited/fieldDelimiter[.='0x20']">,sep=" "
        <xsl:text/>
      </xsl:when>
      <xsl:when test="physical/dataForma/textFormat/simpleDelimited/fieldDelimiter[.='x20']">,sep=" "
        <xsl:text/>
      </xsl:when>
      <xsl:when test="physical/dataFormat/textFormat/simpleDelimited/fieldDelimiter[.='0x09']">,sep="t"
        <xsl:text/>
      </xsl:when>
      <xsl:when test="physical/dataFormat/textFormat/simpleDelimited/fieldDelimiter[.='x09']">,sep="t"
        <xsl:text/>
      </xsl:when>
      <xsl:otherwise> , sep="<xsl:value-of
select="physical/dataFormat/textFormat/simpleDelimited/fieldDelimiter"/>"<xsl:text> </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter[.='']">
      <xsl:choose>
        <xsl:when
          test="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter[.='&quot;&apos;&quot;']">
          , quote="<xsl:value-of select="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter"/>"<xsl:text>
        </xsl:when>
      </xsl:choose>
    </xsl:if>
  </xsl:template>

```

```

        </xsl:when>
        <xsl:when
            test="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter[.='&quot;']">
            , quot='<xsl:value-of select="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter"/>'<xsl:text>
</xsl:text>
        </xsl:when>
        <xsl:otherwise>
            , quot="<xsl:value-of select="physical/dataFormat/textFormat/simpleDelimited/quoteCharacter"/>"<xsl:text>
</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:if>
<xsl:for-each select="attributeList">
    , col.names=c(<xsl:text/>
<xsl:for-each select="attribute">
    <xsl:choose>
        <xsl:when test="position()!<last()>"
            "<xsl:call-template name="attributeName"/>," <xsl:text> </xsl:text>
        </xsl:when>
        <xsl:otherwise>
            "<xsl:call-template name="attributeName"/>" <xsl:text> </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
</xsl:for-each>
    <xsl:text>}</xsl:text>, check.names=T)
</xsl:template>

<xsl:template name="readFixed">
<!--產生一個 FORTRAN 形式的格式 - 但並不允許 tabs -->
tmp_format &lt;- character()<xsl:text></xsl:text>
<xsl:for-each select="attributeList">
    <xsl:for-each select="attribute">
        <xsl:if test="../../physical/dataFormat/textFormat/complex/textFixed[.=' ']">
            <xsl:variable name="nodeNum" select="position( )"/>
            <xsl:variable name="prevNode" select="position( ) - 1"/>
            <xsl:choose>
                <xsl:when test="$nodeNum >1">
                    <xsl:if test="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldStartColumn -
                        ../../physical/dataFormat/textFormat/complex/textFixed[$prevNode]/fieldWidth -
                        ../../physical/dataFormat/textFormat/complex/textFixed[$prevNode]/fieldStartColumn &gt;
0">
tmp_format &lt;- c(tmp_format, "<xsl:value-of
select="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldStartColumn -
                        ../../physical/dataFormat/textFormat/complex/textFixed[$prev
Node]/fieldWidth -

```

```

Node]/fieldStartColumn"/>X")<xsl:text></xsl:text>
    </xsl:if>
  </xsl:when>
  <xsl:otherwise>
    <xsl:if test="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldStartColumn[ . >
1]">
tmp_format &lt;- c(tmp_format, "<xsl:value-of
select="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldStartColumn - 1"/>X")<xsl:text></xsl:text>
    </xsl:if>
  </xsl:otherwise>
</xsl:choose>
<xsl:choose>
  <xsl:when test="(storageType = 'varchar') or (storageType = 'string') or (starts-with(storageType, 'char'))">
tmp_format &lt;- c(tmp_format, "A<xsl:value-of
select="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldWidth "/">")<xsl:text></xsl:text>
    </xsl:when>
  <xsl:when test="(starts-with(storageType, 'ini')) or (storageType = 'byte' )">
    <xsl:choose>
      <xsl:when test="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldWidth[ . >
0]">
tmp_format &lt;- c(tmp_format, "<xsl:text>F</xsl:text><xsl:value-of
select="../../physical/dataFormat/textFormat/complex/textFixed[$nodeNum]/fieldWidth"></xsl:value-
of><xsl:text></xsl:text>
    </xsl:when>
  </xsl:choose>
  </xsl:when>
  <xsl:otherwise>
    <xsl:text> F </xsl:text>
  </xsl:otherwise>
</xsl:choose>
</xsl:if>
  <xsl:text></xsl:text>
</xsl:for-each>
</xsl:for-each>
tmp_format<xsl:text/>

  <xsl:for-each select="attributeList">
tmp_cois &lt;- c(<xsl:text/>
  <xsl:for-each select="attribute">
    <xsl:choose>
      <xsl:when test="position( ) !=last( )">
        "<xsl:value-of select="attributeName"/>", <xsl:text/>
      </xsl:when>
      <xsl:otherwise>
        "<xsl:value-of select="attributeName"/>")<xsl:text/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:for-each>

```

```

    </xsl:choose>
</xsl:for-each>
    </xsl:for-each>
#建一個 data.frame，名爲 dataTable<xsl:value-of select="position( )"/>
dataTable<xsl:value-of select="position( )"/> &lt;- read.fortran(infile<xsl:value-of select="position( )"/>, tmp_format,
col.names=tmp_cols, check.names=T, na.strings=c("NA", ". "))<xsl:text/>
rm(tmp_format, tmp_cols)
</xsl:template>

<!-- 整理 attributeName-->
<xsl:template name="attributeName">
<xsl:text disable-output-escaping="yes"></xsl:text>
<xsl:value-of select="translate( normalize-space(translate(attributeName,'&quot;&amp;'+~%*#@#,:;!^(){}[]|'',' ')),' ','_')"/>

<!-- <xsl:choose>
<xsl:when test="//attributeName[starts-with(., '&quot;')] ">
<xsl:variable name="AtStr1" select="substring-before(substring-after(attributeName, '&quot;'), '&quot;')"/>
<xsl:choose>
<xsl:when test="contains($AtStr1, ' ') ">
<xsl:variable name="AtStr2" select="translate($AtStr1, ' ', '_') />
<xsl:value-of select="$AtStr2" /><xsl:text></xsl:text> :
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$AtStr1" /><xsl:text></xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<xsl:choose>
<xsl:when test="contains(attributeName, ' ') ">
<xsl:variable name="AtStr3" select="translate(attributeName, ' ', '_') />
<xsl:value-of select="$AtStr3" /><xsl:text></xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="attributeName" /><xsl:text></xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
-->
</xsl:template>
</xsl:stylesheet>

```