東海大學統計學研究所

碩士論文

指導教授：魏文翔 教授

統計網路應用與 Linux 之整合及 Java 統計出題介面之建構

Linux Statistical Web-Service and Design of a Statistical Test Using Java

研究生：吳銘陽

中華民國一百年七月

摘要

本文章是將統計與網路做結合，期望透過網路跨平台的特性，來推廣統計方法的應用。在統計方法方面，主要著重在多變量的主成分分析(Principle Component Analysis)以及區別分析(Discriminant Analysis)。另外我們使用昇陽(Sun)公司所提供的 JAVA 軟體 J2SE 來編寫題庫系統，在網路技術方面則是藉由阿帕契(Apache)公司所提供的 Tomcat、Axis 軟體，以及 Linux 系統當中的 CentOS 版本，來作為統計與 Web Service 溝通的平台。希望透過這樣的方式，讓更多人透過網路的管道，更方便的應用統計、推廣統計。而題庫系統也能讓不懂程式的人，輕鬆製作屬於自己的題庫系統。

關鍵字：Axis, CentOS, Discriminant Analysis, JAVA, Linux, Principle Component Analysis, Tomcat, Web Service。

謝誌

在面臨無數次的失敗，我總算可以大聲的說:「我成功了!」這篇文章對我來說，可說是一大挑戰，因為我從完全不懂程式語言，靠著不斷努力，現在總算也漸漸上手，這段過程絕對是我這一生相當難忘的體驗。

我很感謝我的指導教授 魏文翔教授，因為有他的循循善誘，打開了我的視野，讓我的研究生生涯不單單只是與數字打交道，他讓我看到了另一個可能，而我也很努力的創造了一個可能性。感謝張玉媚博士在課程上對我的教誨，她讓我體認到專業知識固然重要，但表達能力與溝通技巧，更加是成功的鎖鑰。感謝楊菁菁博士對我論文的提點，鞭辟入裡的指正與建議讓我受益良多。

最後我要非常感謝我的家人，他們的支持與鼓勵絕對是我面對挑戰的最佳勇氣。

求學路途雖艱辛卻很充實，我相信此刻絕對不是終點，因為往後的人生旅途必定會面臨更多的挑戰與困難，我會謹記老師們的教誨，努力不懈，迎接未知的挑戰。

吳銘陽僅識於

東海大學統計研究所

中華民國一百年七月

目錄

第壹章　緒論

在資訊爆炸的現今，網路技術的應用也越來越純熟，相對的，人們也越來越依賴網路，起初網路只是軍方為了情報蒐集而誕生，然而現在網路卻普及的應用到人們的生活當中，而網路的便利性，也讓現今的人們生活離不開網路了。統計所涵蓋的層面相當的廣泛，舉凡工業、商業、農業甚至醫學，只要需要資料分析的地方，統計便無所不在，而統計除了艱澀的理論外，在分析資料的龐大計算量也讓人相當頭痛，因此若能透過網路的便利性，可望讓統計在應用上更方便，在面對龐大的資料分析，也不用苦於無統計軟體可使用的窘境。相對的，題庫系統也是，以往想要製作題庫都必須消耗大量的人力物力，相當的浪費資源，如今若有一套完善的題庫系統能從旁協助製作題庫，無疑是一大福音。

第一節 研究動機

現今市場上有許多的統計軟體，包括有 SAS、S-PLUS、SPSS、MINITAB 以及 R 語言等，而當中 SPSS、MINITAB 這兩種屬於視窗介面型，在使用上感覺較人性化，也非常方便。而 SAS、S-PLUS、R 語言，

則必須了解程式裡頭相關的語法，才有辦法使用，雖然後三者功能性較強大，但比起前兩者卻是比較難上手的。以上所提及的軟體，除了R語言為免費外，其他都有版權的問題，對於資金不足的人來說又是另一個頭痛的課題。而本篇文章另一大重點乃是題庫系統，傳統的紙本題庫測驗，製作起來不但繁瑣，測驗結束的批閱也相當費時。

本篇研究便是為了解決以上的問題而產生的，我們想透過網際網路當中的Web Service技術，利用其跨程式語言以及程式共享的特性，來提升統計應用的便利性，故此，便以Web Service來當作本篇研究的網路技術。而當中我們又以無版權限制的Linux來當作Web Service的應用平台，除了便利性之外，也兼顧低成本的特性，而因為網路的便利性，也改進了題庫測驗的簡易度，利用JAVA程式語言的幫助結合XML(**Extensible Markup Language**)，讓不懂得撰寫程式的人，也能透過XML的幫助，在不需要了解如何撰寫程式的情形下，快速的製作出專屬的題庫系統，未來更可結合Web Service將題庫測驗變的即時，測驗結束之後，不必再耗費時間等待批閱，透過網路的幫助可以即時的核對答案，不但省去了不少時間，也節省了不少資源。

第二節　研究目的

本篇研究著重於透過網路的便利性來推廣統計方法的使用，並附加題庫系統，不但能讓使用者輕鬆的編寫屬於自己的題庫，也能運用在教學上，讓學習、應用統計變得更加便利。

第三節　章節架構

本研究架構可分為以下幾個部分。第二章軟體探討，探討建立 Web Service 所需使用到的相關軟體。第三章基本觀念，介紹何謂 Web Service 以及它是如何構成的，以及題庫系統編寫的概念。第四章實際應用，主要在於解說建立 Web Service 的流程，及其相關設定。第五章結論，彙整本研究結果及其過程中所遭遇到的困難，還有提出後續研究可進一步補強之處。

第貳章　軟體探討

網路服務的架構十分繁瑣，並非單靠一種軟體就可達成，而是得經由某些特定軟體間相互的整合才能完成。而在使用這些軟體前，對這些軟體有基本的認識是很重要的。底下我們將一一介紹。

第一節 JAVA

Java 是在 1991 年由 James Gosling、Patrick Naughton、Chris Warth、Ed Frank 與 Mike Sheridan 在 Sun Microsystems 公司構思而成。特別的是，Java 最初的動機並不是 Internet，而是為了尋求一個跨平台(也就是與電腦內部結構無關)的電腦語言，讓此語言所建立的軟體可嵌入各種不同的消費性電子裝置。為了達到此目的，Gosling 與其它人開始建立可移植、跨平台的語言，因此創立了 Java。而 Java 也變成現今重要的程式語言之一。

一、特性

1. 安全性

當下載程式時，都有感染病毒的危險。除了病毒之外，還必須防範另一種具有惡意的程式。此種程式可以藉著蒐尋你電腦檔案系統中的內容來取得私人的資訊，而 Java 藉著在網路應用程式與你的電腦之間提供「防火牆」來排除這些憂慮。當你使用 Java 相容的 Web 瀏覽器時，你可以安全的下載 Java applet，無須擔心病毒或惡意的程式。Java 將 Java 程式侷限在 Java 執行環境下，並且不容許它存取電腦的其它部分來達成這項保護。無須擔心會有其它的傷害以及侵入的安全問題，這被認為是 Java 最重要的部分。

2. 移植性

Java 的第二個動機就是 World Wide Web。在 1993 年，Java 設計團隊發覺，當企圖建立 Internet 程式碼時，也時常會碰上為嵌入式控制器建立程式時相同的移植問題。世界各地使用各種不同型式的電腦和作業系統，其中一大部分都可以連接到 Internet。對於那些動態下載到 Internet 上各種平台的程式而言，必須要有一些方法來產生可移植的程式碼，用來確保安全性的機制同樣也可用來建立可移植的程式，而 Java 對這兩個問題的解決方法確實很精簡，而且很有效率，Java 設計者最主要的考量是程式碼的壽命和移植性。程式設計師所面臨的主要問題是，今天所寫的程式並不保證明天還可以執行，即便是在相同的機器上。作業系統的升級、處理器的升級以及核

心系統資源的更動都可能讓程式的功能失常。而 Java 的目標便是”write once; run anywhere, any time, forever.”

3. 物件導向

在早期，對於會重複使用的程式都是一寫再寫也就是當某一程式碼需要應用在不同的程式時，那麼這程式碼得再寫第二次，所寫的內容與上次是大略相同，若只是簡單的程式可以一直重複寫，然而對於困難或複雜的程式而言，重複寫就變成了一個苦差事，並且所寫出來的程式必定會很冗長。物件導向程式的特色就是將常會使用到的程式碼組合起來，形成一個「物件」，只須寫一次程式碼就可以應用在其他程式上，無須在寫第二次。物件導向程式設計的組織方式是圍繞著資料(也就是物件)和一套定義良好的介面，物件導向程式的特徵在於「資料控制程式碼的存取」，Java中的物件模型很簡單，且容易擴展。只有像整數這樣的簡單型態才是以非物件的方式來處理。

4. 自動垃圾回收

C 語言中動態配置的物件必須以手動的方式釋放，而 Java 採用不同的方法，它自動的處理解除配置，完成這項工作的技術稱為垃圾收集。當沒有任何參考指向物件時，Java 就假設此物件不再需要了，所以此物件所佔據的記憶體可以再回收利用。而在程式執行期間，垃圾收集只會偶爾出現，不會因為一兩個物件不再使用就出現進行垃圾

收集。此外，不同的 Java 執行期間實作會採取不同的方式來作垃圾收集，重要的是，在寫程式的過程不需要考慮它。

5. 例外處理

例外是一種不正常的情況，在執行期間由程式碼產生。換句話說，例外是一種執行時期的錯誤。在沒有支援例外的電腦語言裡，錯誤必須要人工檢查與處理，這種處理方式很累贅而且很麻煩。Java 的例外處理可以避免這些問題，而且在處理過程中，將執行時期的錯誤管理帶進物件導向的世界。

二、版本

依據使用者不同的需求，Java 也提供了不同的版本，如下所示：

1. J2SE (Java 2 Platform ，Standard Edition)

Java 的標準版本，若要開始學Java 程式語言時，都是從這個版本開始學習的。在這文章中，所使用的就是這個版本。

2. J2EE (Java 2 Platform ，Enterprise Edition)

Java 的企業版本，針對企業機構而開發，然而在網際網路上比 J2SE 還要廣。

3. J2ME (Java 2 Platform ，Micro Edition)

Java 最精簡的版本，通常是應用在電子方面，例如手機、Java Card

等。

第二節　Apache

1995 年以前就有很多的 WWW 伺服器軟體，其中以 HTTPd 佔有率較高。 後來 HTTPd 經過多次臭蟲的修訂後，才在 1995 年後發佈 Apache (A patche server) ，主要提供 WWW 的伺服器平台。全世界幾乎大型的 Web Site 都是採用 Apache 作為 WWW Server，無論是 Uinx、Sun Solaris、Free BSD、Linux... 系統清一色都是以 Apache 架設 Web Server，甚至 NT 系統也有不少以 Apache 架設而成的 Web Server。

Apache 是 Linux 系統上最廣泛用來架設 WWW 伺服器的架站軟體，由 Apache Group 在開發維護，1999 年 6 月 30 日成立 Apache Software Foundation。從原本的業餘玩家所組成的 Apache Group，到現在有幾家主要的公司如 IBM、Sun Microsystem、Oracle 等在背後支持後續的計畫，使得 Apache 伺服器可以持續的發展。

第三節 Apache Tomcat

Apache Tomcat 是一個開放原始碼的 Java Servlet and JavaServer Pages 補充程式。Java Servlet and JavaServer Pages 的規格由 Java 社群開發的。Apache Tomcat 的開發與參與是在 Apache 軟體授權之下。在網頁應用伺服器上，Apache 為眾多伺服器的首選，對於 JSP 的服務來說當然是以 Tomcat 為主要的選擇，在目前的主流上若要使用 JSP 開發 Web 網頁程式的話，Apache 和 Tomcat 當然是最佳選擇。

第四節 Apache Axis

Axis為Apache軟體基金所開發，它專門提供Web Service的服務架構，通常需與Apache Tomcat一起整合，成為一個完整的Web Service。除了使用Axis作為Web Service中所需的軟體之一，亦可以使用Soap軟體，其實Axis與Soap是同一性質的，都是Apache軟體基金開發，以及都是有提供Web Service服務架構，然而它們的差別只在於版本的問題，Soap可以說是Axis的前一代，因為在一開始Soap被發現有些地方有缺陷，因此Apache軟體基金沒有修改Soap，反倒是重新

編寫了新的軟體，為了與Soap有所區別，因此命名為Axis。

第五節 Linux CentOS版本

早在Linux出現之前的二十年(大約在1970 年代)，就有一個相當穩定而成熟的作業系統存在了！ 那就是『Unix』！眾所皆知的，Linux的核心是由Linus Torvalds在1991年的時候開發出來的， 並且放到網路上提供大家下載，後來大家覺得這個小東西(Linux Kernel)相當的小而精巧， 所以慢慢的就有相當多的朋友投入這個小東西的研究領域裡面去了！而之所以取名為 Linux 是因為 Torvalds 放置核心的那個FTP網站的目錄為：Linux， 從此，大家便稱這個核心為Linux 了。不過，由於發展Linux distributions的社群與公司實在太多了，例如在台灣有名的 Red Hat, SuSE, Ubuntu, Fedora, Debian 等等，所以很多人都很擔心，如此一來每個distribution是否都不相同呢？因為每個Linux distributions使用的kernel都是http://www.kernel.org所釋出的，而他們所選擇的軟體，幾乎都是目前很知名的軟體，重複性相當的高，例如網頁伺服器的 Apache，電子郵件伺服器的 Postfix/sendmail，檔案伺服器的 Samba 等等。此外，為了讓所有的Linux distributions開發不致於差異太大，且

讓這些開發商在開發的時候有所依據，還有 Linux Standard Base (LSB)等標準來規範開發者，以及目錄架構的 File System Hierarchy Standard (FHS)標準規範！ 唯一差別的，可能就是該開發者自家所開發出來的管理工具，以及套件管理的模式吧！基本上，每個 Linux distributions 除了架構的嚴謹度與選擇的套件內容外， 其實差異並不太大！因此，大家可以選擇自己喜好的 distribution 來安裝即可。在此篇論文，我使用的是號稱完全相容商業版 RHEL 的 CentOS。

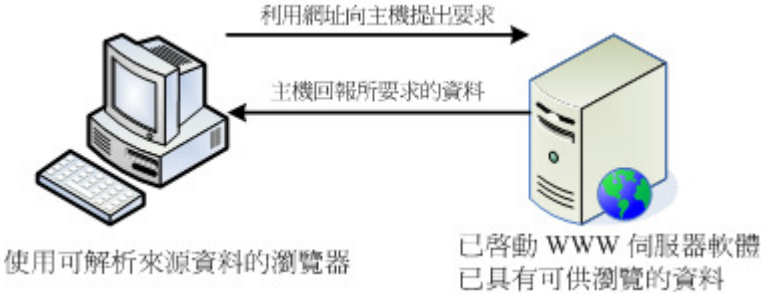
第參章　基本觀念

第一節 Web Service

目前網際網路社會上最熱門的協定就是 TCP/IP 了，而 TCP/IP 的應用方面則有 mail, dns, ftp, telnet 與 www 等等，其中造成 TCP/IP 大流行的應屬 mail 與 www，尤其是 WWW。 WWW 是 Word Wide Web 的縮寫，其中 Web 有廣播網的意思存在， 所以簡稱為全球資訊網。WWW 可以結合文字、圖形、影像以及聲音等多媒體， 並透過超連結 (Hyper Text) 的方式將資訊以 Internet 傳遞到世界

各處去。

與其他的伺服器類似的，你要連結上 WWW 網站時，該網站必需要提供一些資料， 而你的用戶端則必需要使用可以解析這些資料的軟體來處理，那就是瀏覽器。簡單來說，我們可以用下圖來顯示 WWW server/client 的相關性：

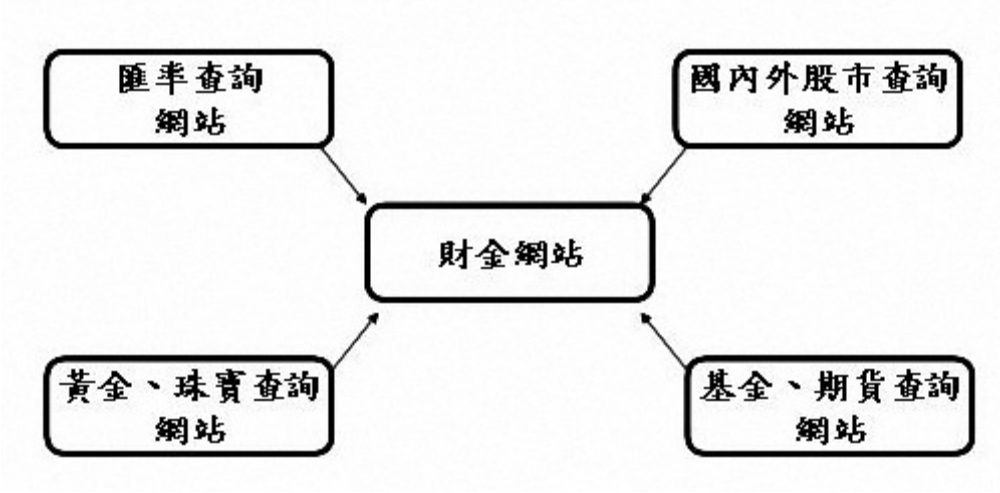


也就是說，你的 WWW 伺服器不但需要一個可讓用戶端瀏覽器瀏覽的平台，還需要提供用戶端一些資料才行！ 那這些可供查詢的資料有哪些類型？最主要的其實是超文件標籤語言（Hyper Text Markup Language, HTML）。 其實超文件標籤語言大多只是一些純文字資料，然而透過瀏覽器對於一些標籤（<tag>）的解釋後， 就能夠得到不同的文字格式、顏色顯示等等。

在過去 Web Service 可以說是網頁的代名詞，經過多年的發展，Web Service 的功能已經超出我們的想像，而當中所涉及到的知識更是無遠弗屆，而當中最基本的部分便是顧客端與伺服器端的連結，新一代的分散式運算技術---「Web Services」就此誕生。

舉例來說，有一個關於財金方面的網站，他提供了四種服務，如

下圖所示：



使用者如果想使用這四種服務的任一種，都可以透過這個財金網站的幫助來連結，而不需要分別去找尋這四個網站並且流覽他們。而當中像是匯率查詢的部分，使用者的電腦並不參與任何的運算過程，而是由匯率查詢網站來協助換算，因此透過財金網站這個平台可以讓大家輕鬆的使用這四種服務。

網際網路標準組織W3C（http://www.w3.org）對Web Service所制定的定義就是Web Service 最基本是從一個應用程式所構成的，然而這一個應用程式要如何在網際網路上提供服務呢?目前很受歡迎的四種技術便是XML、SOAP、WSDL 以及UDDI，若具備這四種技術就可以稱為Web Service，當然這並非絕對，這只是目前受歡迎的做法。底下就針對這四種技術做簡略的介紹。

一、XML(Extensible Markup Language)；可擴展標記語言

標記語言(Markup Language)在這個資訊社會是最被廣泛使用的語言之一，一個使用標記語言的例子，便是與我們日常生活息息相關的網際網路。在網路上，用來撰寫傳遞資訊網頁的便是標記語言HTML(Hypertext Markup Language)。然而 HTML 並不允許自行去定義標籤(tag)以達到交換資訊的目的，即 HTML 是不可延伸撰寫，人們只能使用 HTML 所提供之標籤來設計網頁，對欲以在標記語言中定義新標籤，來達到資料交換或資料儲存為目標的各行各業，是極不方便的。XML 是從 1996 年開始有其雛形，並向 W3C（全球資訊網聯盟）提案，而在 1998 二月發佈為 W3C 的標準（XML1.0）。XML 的前身是 **SGML**（The **S**tandard **G**eneralized **M**arkup **L**anguage），是自 IBM 從 1960 年代就開始發展 **GML**（**G**eneralized **M**arkup **L**anguage）標準化後的名稱。**可擴展置標語言**（Extensible Markup Language，簡稱 XML），又稱**可擴展標記語言**，是一種置標語言。置標指電腦所能理解的信息符號，通過此種標記，電腦之間可以處理包含各種信息的文章等。如何定義這些標記，既可以選擇國際通用的標記語言，比如 HTML，也可以使用像 XML 這樣由相關人士自由決定的標記語言，這就是語言的可擴展性。XML 是從標準通用置標語言（SGML）中簡化修改出來的。它主要用到的有可擴展置標語言、可擴展樣式語言(XSL)、XBRL 和 XPath

等。

二、SOAP(Simple Object Access Protocol);簡單物件存取協定

SOAP「簡單物件存取協定」，於 2000 年由 IBM、Microsoft、UserLand 和 DevelopMentor 共同提交給 W3C。SOAP 是一個協定規範，定義傳遞 XML 資料的方法及各式各樣的傳輸協定，如 HTTP、FTP、SMTP、TCP 等，SOAP 以 XML 作為資料傳輸的格式，再搭配上述的傳輸協定來傳送訊息。由於 SOAP 具有跨平台的優勢，不但是業界所支援的公開標準，更因其有穿透防火牆的優點，使得遠端程序呼叫(Remote Procedure Call , RPC）可以順利橫跨不同的網域。而在本篇論文當中，我所使用的通訊協定為 HTTP。

三、WSDL(Web Service Description Language);網路服務描述語言

WSDL（Web Service Description Language）在網路服務中，扮演一個重要的角色:『描述網路服務』。如果沒有統一『描述網路服務』的語言，將無法整合服務，WSDL 標準目前由 IBM 及 Microsoft 所研擬，裡面說明相關服務的 IDL（Interface Description Language）及程式間如何協同合作進行網路服務。

WSDL 是利用 XML 語言所寫成，只是副檔名為「.WSDL」。WSDL 語法分為兩個部分：

1. 描述服務介面（Service Interface Definition）

這個部分主要是說明，「.WSDL」檔案資料的格式、傳遞溝通的訊息、如何進行作業等，簡單規範性的說明，這個部分是可以重複使用的。

2. 實做服務定義（Service Implementation Definition）

這個部分主要是說明，由如何進行由服務介面所提供的服務，包括：提供服務的廠商、提供服務的網路位置等

整個 WSDL 的文件是由< definitions >作為整份文件的根元素，<definitions >下，再分為<types>、<message>、<portType>、<binding>、<Service>五個子元素。

通常只要獲得了某一個服務的WSDL 檔，那麼就如同已經獲得到關於這個服務的執行細節與權力。

四、UDDI(Universal Description Discovery and Integration);

通用描述、發現及整合協定

2000 年由微軟、IBM 與 Ariba 提出。理論上，在這套目錄裡註冊過的公司可將此系統當作網路搜尋引擎的商業服務版。例如，一個電子商務網站可利用此目錄資料庫尋找提供信用卡交易機制的公司。若有找到符合的，所有交易過程中的元件(包括售價與付款)都可直接透

過電子交易完成。贊成此計畫的人士表示，將所有的溝通搬上網路可大幅簡化流程並加速生意的成交。協助設立 UDDI 規格的微軟、IBM、SAP 與 HP 目前都有架設網站供網路服務註冊於公共 UDDI 目錄資料庫中。日本的 NTT 電訊公司目前也有一個類似的網站正在架構中。理論上，網路服務接受度越高，軟體與硬體的銷售也會隨著提升。UDDI 是組成網路服務的四大基本技術，其他三大技術包括 XML(延伸標記語言)、SOAP(簡單物件存取協定)與 WSDL(網路服務描述語言)。至於它們之間的連結，我們不妨以打電話來作比喻：在網路服務的世界裡，XML 代表雙方交談的語言，SOAP 則是規定撥打電話的規則，UDDI 則是電話簿，WSDL 則是規範每通電話的性質以及如何參與。

第二節　題庫系統

本篇文章是利用 JAVA 程式語言來撰寫題庫系統，選取 JAVA 程式語言的原因在於它的跨平台特性。題庫系統主要分成三大部分，單選題、複選題、填空題，在做完測驗之後，除了顯示出分數之外，還會顯示出正確的答案以及測驗者的答案提供核對，此題庫系統不同的地方在於結合了 XML(Extensible Markup Language) 可擴展標記語言，讓使用者在編寫題庫的時候，並不涉及到程式語言的部分，換句話

說，就算不懂 JAVA 程式語言，透過 XML 與範例的幫助，也可以很輕鬆的製作出屬於自己的題庫系統。

(1)標籤:

針對題庫系統，本論為定義一些標籤來幫助使用者建立題庫。分別介紹如下:

<testBankSubject>:用來設定題庫主題

<section>:為<testBanksubject>的子標籤,用來設定題庫的單元主題。

<problem>:為<section>的子標籤，為最主要的標籤，用來設定問題的主題及其屬性(單選、複選或填充)包含以下的子標籤:

<text>:用來描述問題的本文。

<item>:用來描述問題的選項。

<solution>:用來給定解答。

<score>:用來給定問題之分數。

以下為一分數為 10 分且答案為第三選項之單選題範例,此題庫主題為”Basic Statistics”而此問題是在”Section1.1”的第”1”題。其型式為”choice”(選擇題)

範例:

```
<?xml version="1.0"?>
```

```
<testBankSubject title="Basic Statistics">
  <section title="Section 1.1">
    <problem title="1" type="choice">
      <text> Quantitative data </text>
      <item> are always nonnumeric </item>
      <item> may be either numeric or nonnumeric </item>
      <item> are always numeric </item>
      <item> are always labels </item>
      <solution> 3 </solution>
      <score> 10 </score>
    </problem>
  </section>
</testBankSubject>
```

(2)DTD(Document Type Definition)語法檢查:

一份良好的 XML 文件，其元素架構、元素標籤以及屬性必須定義良好，如此一來，撰寫者才能依據此定義來產生 XML 文件，並進行驗證。而扮演此良好定義角色的便是 DTD。DTD 可以依附於 XML 文件內，亦可以獨立成為外部檔案，並可置於任何網路可存取的相關伺服器，此題庫系統的 DTD 如下:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT testBankSubject ( section ) >
<!ATTLIST testBankSubject title NMTOKENS #REQUIRED >
<!ELEMENT section ( problem )+ >
<!ATTLIST section title NMTOKENS #REQUIRED >
<!ELEMENT problem ( text, item*, solution+, score+ ) >
<!ATTLIST problem type CDATA #REQUIRED >
<!ATTLIST problem title NMTOKENS #REQUIRED >
<!ELEMENT text ( #PCDATA ) >
<!ELEMENT item ( #PCDATA ) >
<!ELEMENT solution ( #PCDATA ) >
    <!ELEMENT score ( #PCDATA ) >
```

上述 DTD 包含主要兩種標籤，<!ELEMENT>及<!ATTLIST>，分別用來設定 XML 文件的標籤組成成分(Element)，以及其相關屬性(Attribute)，符號”*”及”+”設定子元素出現次數，”+”代表此子元素至少出現一次，而”*”代表此子元素可出現零或多次，而 CDATA 及 NMTOKENS 指出此元素的屬性型態，CDATA 為一般文字資料的字串，而 NMTOKENS 為一般文字資料，可包含字母及數字，且可用空白分隔多組，而其後之”#REQUIRED”代表

此屬性為必要，最後 PCDATA 代表文件的文字內容，例如在上小節範例” Quantitative data”是元素 text 的 PCDATA。

第肆章 實做應用

第一節、題庫系統

所需使用的軟體：JDK-1.6.0_14

Step 1.安裝 JDK 程式

Java 軟體可在昇陽(Sun Microsystems)公司專屬網站 http://java.sun.com下載，下載時請特別注意選用『Linux』版本，安裝方式分為兩種，一種是以圖形介面的方式安裝，執行方式與 Windows作業系統雷同，另一種方式為文字介面方式，在此以文字介面來說明安裝過程。首先，開啟終端機，利用『cd』指令切換至存放安裝檔的目錄，例如我將安裝檔存放在『/opt/java』目錄底下，因此我可以輸入以下指令切換到該目錄：

```
[root@localhost ~]#cd /usr/local/java
```

接著在命令列輸入以下指令：

```
[root@localhost ~]#./jdk-6u20-linux-i586-rpm.bin
```

其中『.』代表的是當前的目錄，而『/jdk-6u20-linux-i586-rpm.bin』則為該安裝檔檔名。

Step 2.設定環境變數

環境變數的設定可在 /etc/profile 設定，在命令列輸入『vi /etc/profile』，利用 vi 這個內設的文字編輯器來開啟 profile。並在文件最下方輸入以下文字：

```
export JAVA_HOME=/usr/local/java
export PATH=$JAVA_HOME/bin:$PATH
```

而關於vi編輯器的使用方式可以參照附錄A。

設定完成後可使用『source』指令或『.』指令讓修改的項目生效。

```
[root@localhost ~]# . /etc/profile
```

如此，便可開始使用 Java 來撰寫程式。

Step 3.題庫系統解說

題庫系統主要分成三大部分，分別是單選題、複選題以及填空題，而相關的 Java 程式碼請參照附錄 B。

首先，在執行題庫系統時，會開啟以下的視窗(如<圖一>)：

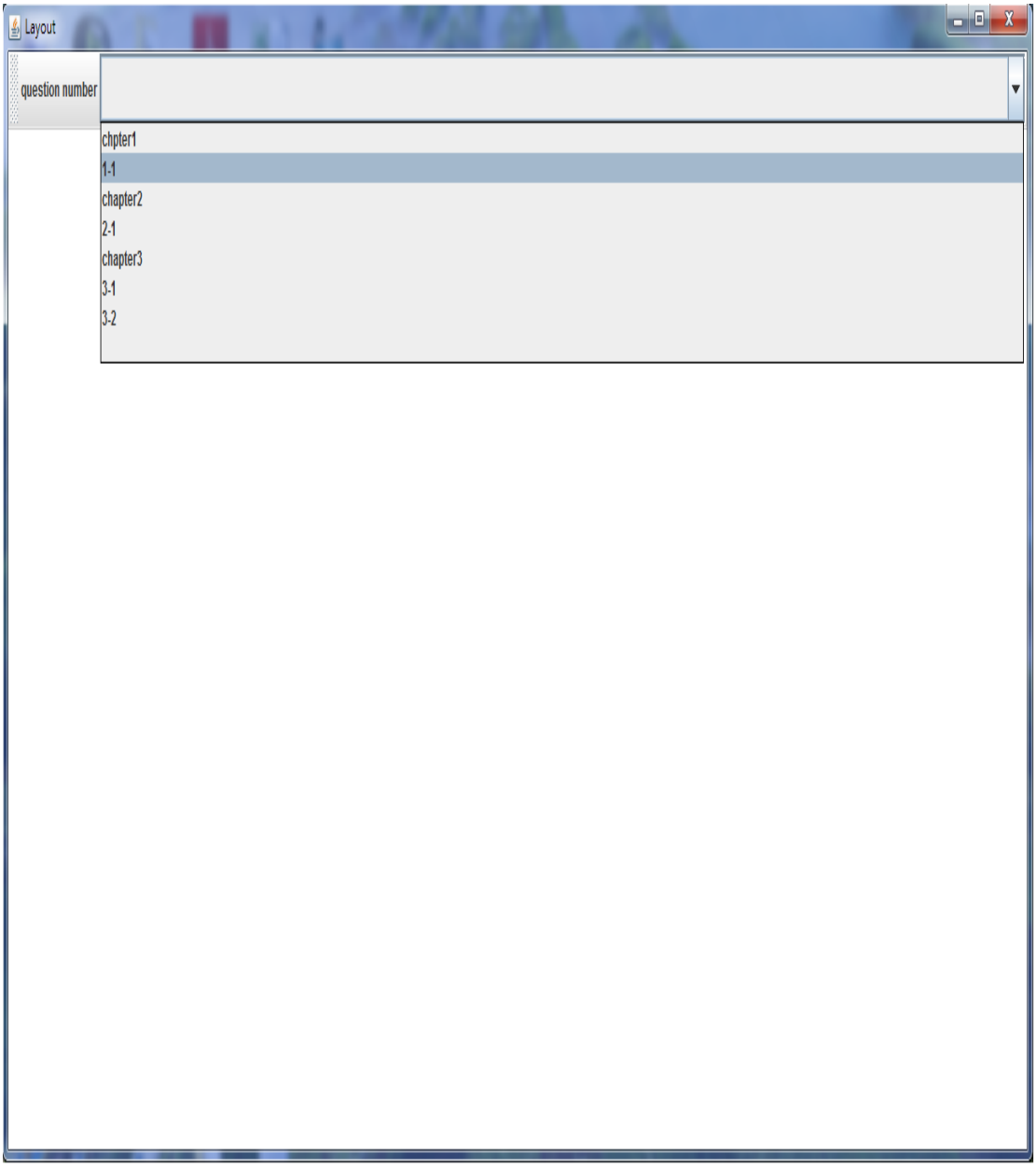


figure 1

<圖一>的 question number 標籤旁提供了一個下拉式的卷軸，我們可以看見當中有 chapter1、1-1、chapter2 等等，分別代表的是章節，這個部分使用者也可以透過 XML 的協助輕鬆的更改為自己想要的名稱，而在下拉式捲軸當中選取希望測驗的章節便會開始進入測驗，底下將以 chapter1 的 1-1 為範例來說明。

1. 單選題

首先進入的為單選題，使用者可以透過 XML 的協助設定題目、配分、選項內容以及設定標準答案的索引號碼，而當測驗者選取預作答的選項時，畫面會出現確認窗口，詢問測驗者是否確定選取該選項，若測驗者選擇『否(N)』，則會繼續停留在該題等待測驗者重新作答，若選擇『是(Y)』系統便會儲存測驗者的答案並進行下一題。

2. 複選題

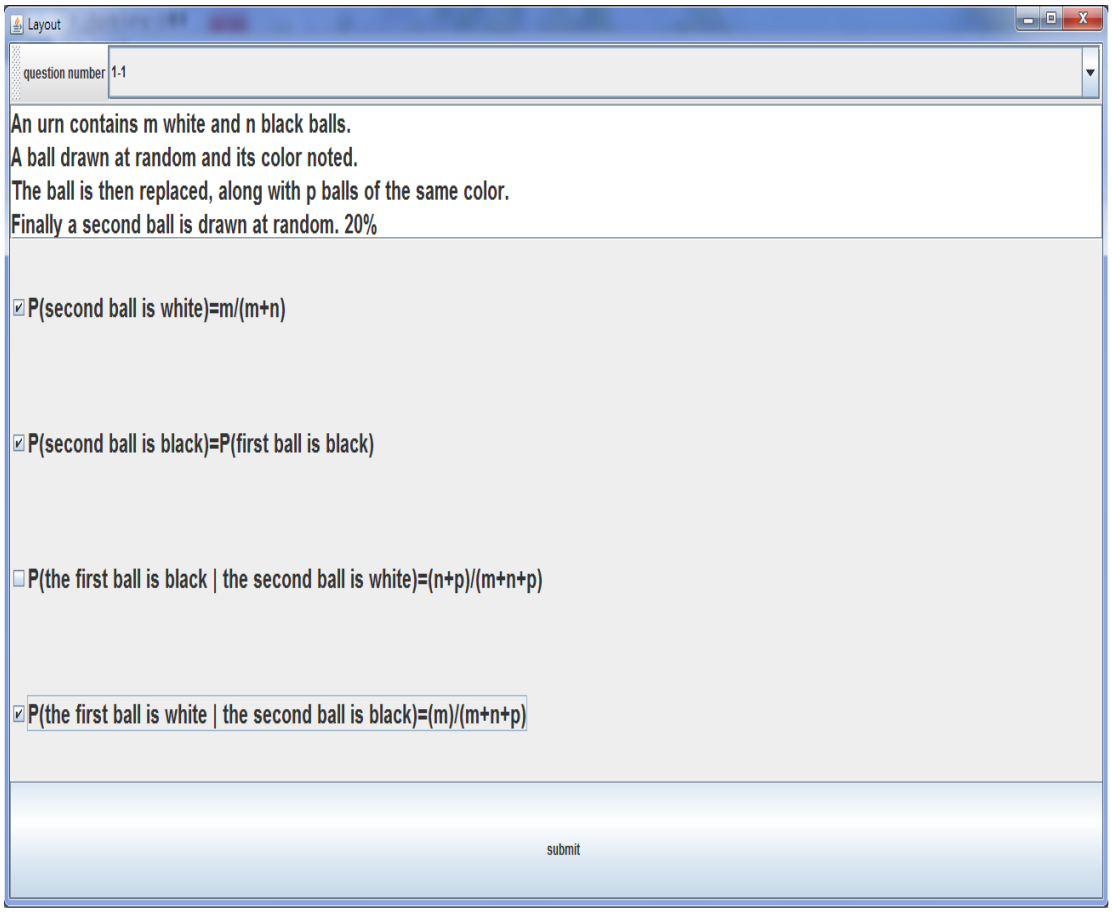


figure 2

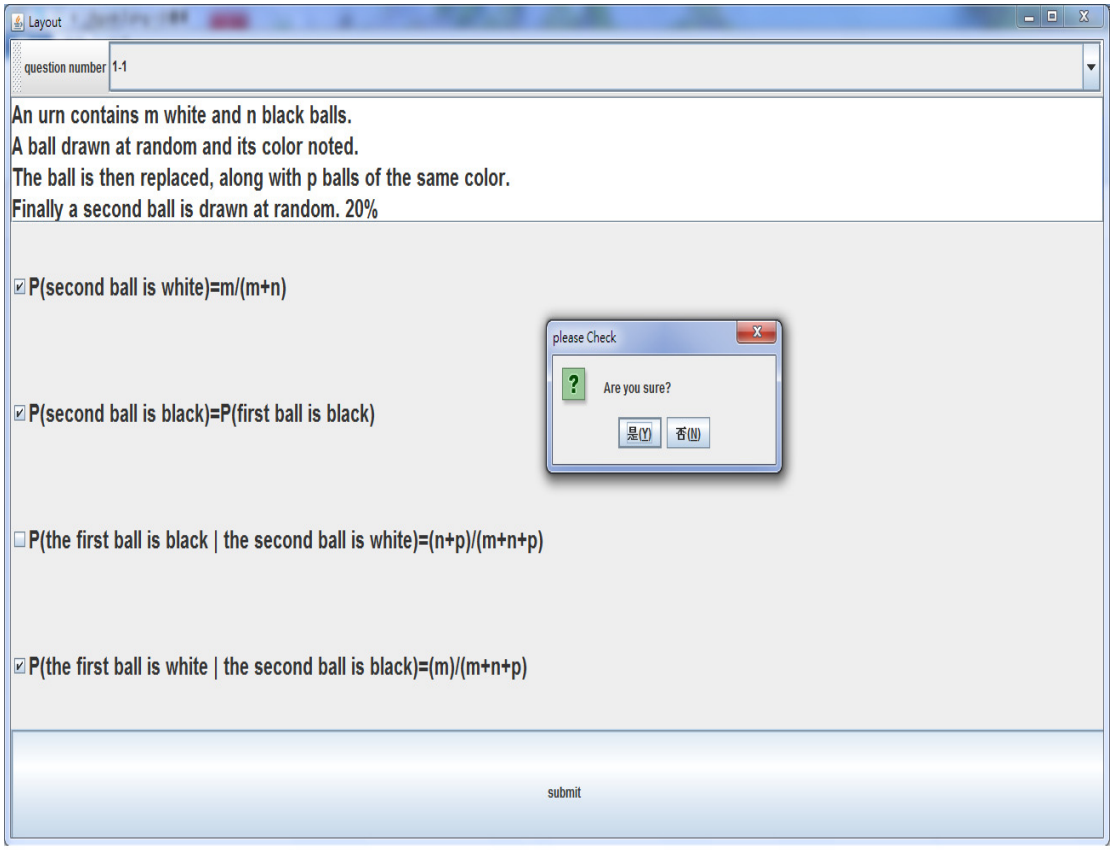


figure 3

圖 2 為複選題的視窗，與單選題相同的是，使用者可以透過 XML 的協助設定題目、配分、選項內容以及設定標準答案的索引號碼，而此處因為是複選題，所以設定的答案索引號碼會超過一個，我們也可以發現在視窗底部，比起單選題的視窗多了一個按鍵，在測驗者選取了期望作答的選項之後，透過底部按鈕的幫忙，來達到送出結果的動作，這是為了因應複選題而產生的作法，與單選題相同的，在送出結果之後，畫面會開啟確認視窗(如<圖 3>)，詢問測驗者是否確定選取該選項，若測驗者選擇『否(N)』，則會繼續停留在該題等待測驗者重新作答，若選擇『是(Y)』系統便會儲存測驗者的答案並進行下一題。

3.填空題

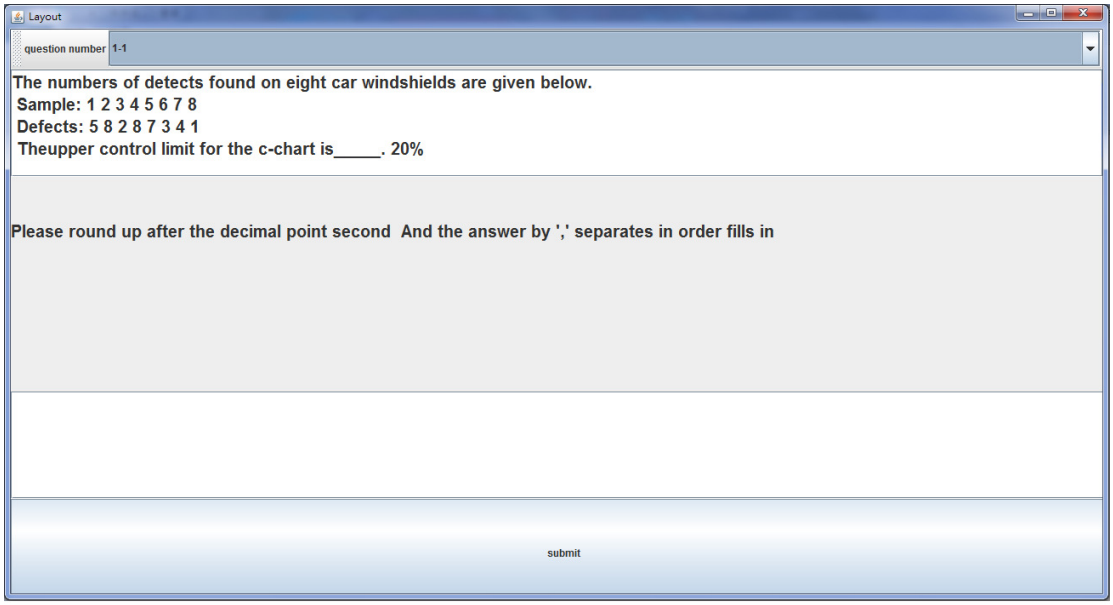


figure 4

圖 4 為填空題的視窗，使用者可以透過 XML 的協助設定題目、配分以及設定標準答案，值得注意的是此部分為填空題，因此答案是採用字串方式儲存，當使用者在設定答案時要以『,』隔開子小題的個別答案，而視窗中間區塊也以文字提示測驗者，在作答時要以『,』隔開子小題的個別答案，填空題視窗底部的按鈕，為提供送出結果的動作，在測驗者完成了答案的填寫之後，透過底部按鈕的幫忙，來達到送出結果的動作，在送出結果之後，畫面會開啟確認視窗，詢問測驗者是否確定送出該結果，若測驗者選擇『否(N)』，則會繼續停留在該題等待測驗者重新作答，若選擇『是(Y)』系統便會儲存測驗者的答案並進行下一題。

值得注意的是，如果測驗者輸入的答案數目和題目數目不同時，系統會給予警告訊息，並在測驗者更改為相同數目之後，才能繼續進行下一題。

4. 核對

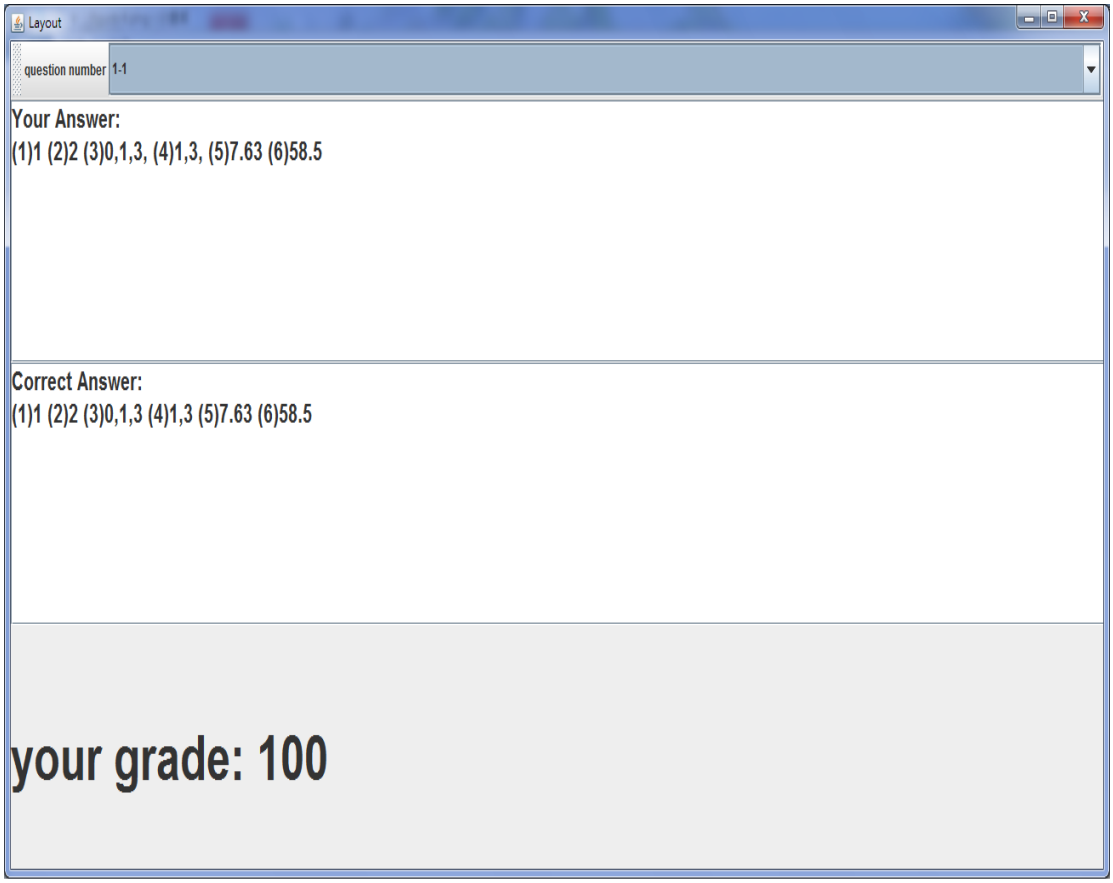


figure 5

在測驗者作答結束之後，系統會自動開始核對的視窗，此階段分為三個部分，第一個部分顯示出測驗者的所有答案，第二部分顯示出使用者所設定的標準答案，第三個部分則為依據使用者在每一題設定的配分，在核對測驗者的答案之後，所給予的成績。

第二節、Web Service

所需使用軟體及套件:

1. JDK-1.6.0_14
2. apache-tomcat-5.5.33.tar.gz
3. axis-bin-1_4.tar.gz
4. activation.jar、mail.jar、xmlsec.jar、

Step 1.安裝 JDK 程式

關於此步驟的相關安裝設定，可參考上一節的解說，此處便不再說明。

Step 2.安裝 Apache Tomcat

軟體可在 http://www.apache.org 網址下載，安裝檔有分為 Binary Distributions 以及 Source Code Distributions 兩種，前者為已經編譯好可以使用的,後者為可修改當中程式碼並編譯成符合自己需求。在此，我們使用前者。將安裝檔下載至桌面，對其點選右鍵解壓縮，然後開啟終端機輸入以下指令:

```
[root@localhost ~]#cd /root/Desktop
[root@localhost ~]#mv apache-tomcat-5.5.33 /opt/tomcat
```

安裝完畢之後，必須要設定相對的環境變數，才能順利啟動 Tomcat，有關 Tomcat 的環境變數設定可以參考附錄 B。

在 linux 底下要將 Tomcat 啟動十分簡單，只需要在終端機輸入以下指令:

```
[root@localhost ~]#/opt/tomcat/bin/startup.sh
```

可得到以下輸出:

```
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/local/java
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar
```

若沒辦法順利啟動，即有可能為檔案權限問題，因此可輸入以下指令來達到修改權限的目的:

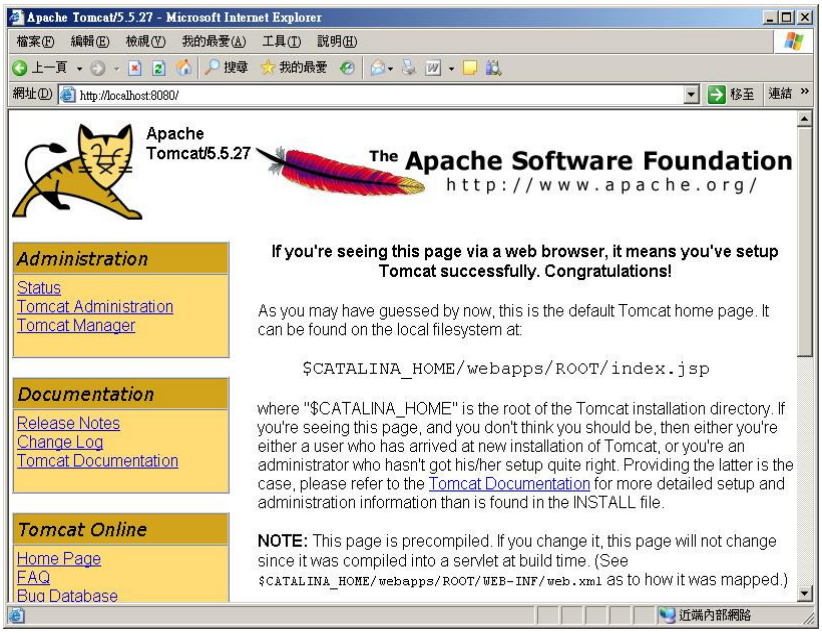
```
[root@localhost ~]#chmod 755 /opt/tomcat/bin/startup.sh
[root@localhost ~]#chmod 755 /opt/tomcat/bin/shutdown.sh
```

可以輸入以下指令確認 Tomcat 是否有啟動:

```
[root@localhost ~]# netstat -nult | grep :80

tcp 0 0 :::8080  :::*    LISTEN
```

其中 8080 便是 Tomcat 預設啟動的埠口。輸入網址：http://localhost:8080，便會出現<圖 1>的畫面



<圖 1>Tomcat 主畫面

若沒有辦法順利連線，有可能是防火牆阻擋所造成，只要關閉防火牆便可順利連線。因此可以輸入以下指令來關閉防火牆:

```
[root@localhost ~]#/etc/init.d/iptables stop
```

若要關閉 Tomcat，只需要輸入以下指令便可完成:

```
[root@localhost ~]#/opt/tomcat/bin/shutdown.sh
```

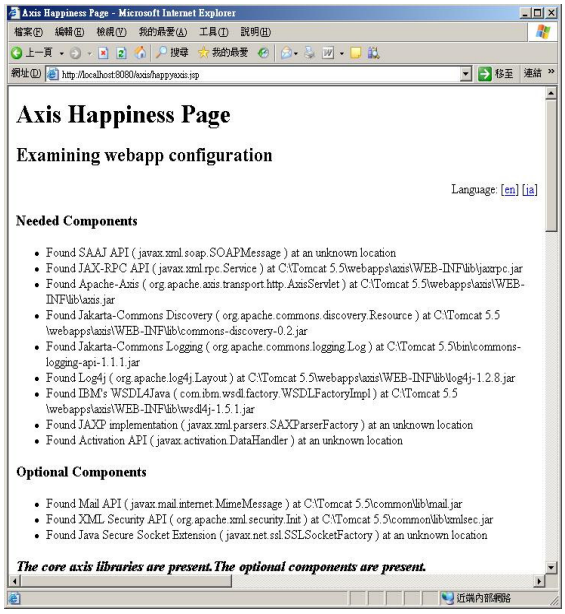
Step 3.將 AXIS 1.4 解壓縮

可到http://www.apache.org 網址下載其壓縮檔，將安裝檔下載至桌面，對其點選右鍵解壓縮，然後開啟終端機輸入以下指令:

```
[root@localhost ~]#cd /root/Desktop
[root@localhost ~]#mv axis-1_4 /opt/axis
```

在/opt/axis/webapps 目錄裡有一個檔名為axis 的目錄，將它複製到之前Tomcat 所安裝路徑/opt/tomcat/webapps 目錄內，並且將相關的API:activation.jar、mail.jar、xmlsec.jar、xml-apis.jar 以及javastatsoft_beta1.3.jar 放置/opt/tomcat/common/lib 目錄中，其中前面四個API是為了在使用AXIS時不發生錯誤而必須添加的，可以在昇陽(Sun Microsystems)公司網頁下載，而第五個就是統計相關的API。之後必須將在所有放置在/opt/tomcat/common/lib 目錄中的jar 檔，都設定Classpath 的環境變數，細節參考附錄B。在這所提到的「API(Application Programming Interface)」就是一組其他程式設計師寫好的程式，只須要依照它的使用規則，就可以順利使用它，有點類似於在R 語言中的套件。輸入網址http://localhost:8080/axis，出現<圖2>，然後若要驗證其API 是否有缺少時，可以執行Validation 來確定是否有發生錯誤，如<圖3>為正確的畫面。

<圖2>Apache Axis 主畫面



<圖3>驗證是否有發生錯誤

Step 4. 撰寫服務程式並產生WSDL 檔

在論文當中，所要提供的服務就是在統計非常實用的多變量分析(陳順宇,2000)，在此將提供主成分分析以及區別分析，利用關於統計的API-javastatsoft_beta1.3.jar，將服務程式完成，其程式碼請見附錄C。利用主成分分析與區別分析的程式碼，來產生WSDL 檔，其分別有兩種方法可使用，其程式碼是相同的，只是所存的副檔名不同，分別如下:

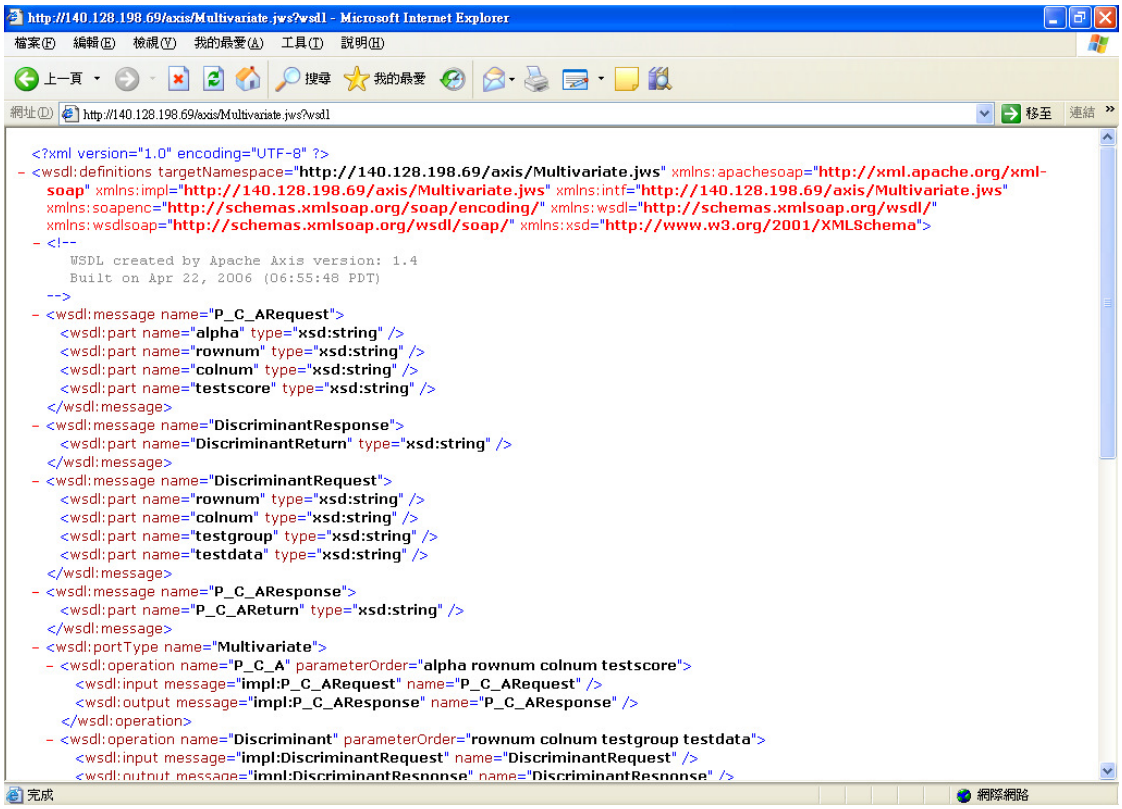
第一種：使用JWS 檔

將所需寫的程式碼完成於記事本上，這時存檔存成.jws，放置/opt/tomcat/webapps/axis 目錄內，撰寫服務程式，命名為Multivariate.jws，代表著多變量分析，而程式碼當中包含有主成分

分析以及區別分析。

假如要查看的服務是多變量分析時，輸入網址http://localhost:8080/axis/Multivariate.jws，就會出現<圖4>的畫面，再執行Click to seethe WSDL 時，就會產生關於服務程式的WSDL 檔，或者是在上述的網址後面多打上?wsdl，產生WSDL 檔，如<圖5>。

<圖4>服務畫面

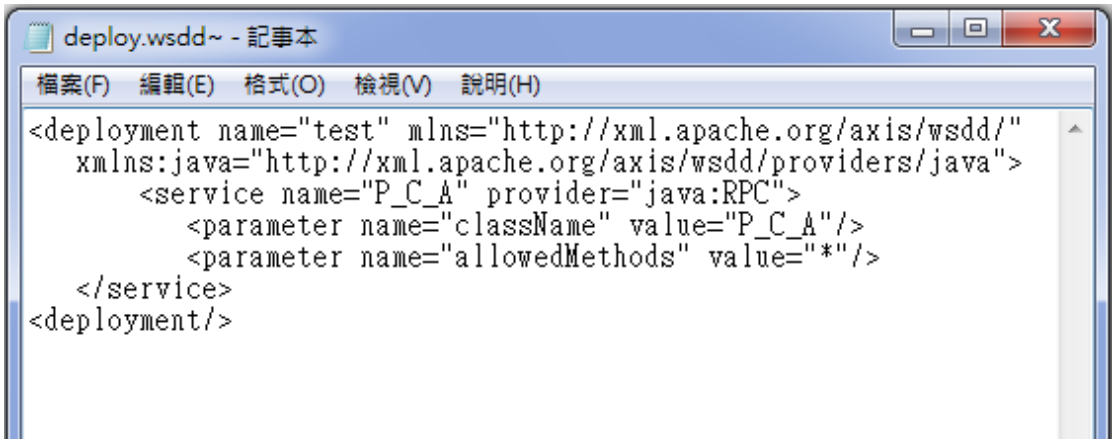


<圖5>服務程式的WSDL檔

從所產生的WSDL檔<圖5>，在最上方第一段就是在說明這Web Services名稱及位址等資訊，在第二段就是在描述有哪些參數、參數名稱及屬性。

第二種：使用CLASS 檔

將所需寫的程式碼完成於記事本上，這時存檔存成java 檔，經由編譯所產生的CLASS 檔放置/opt/tomcat/webapps/axis/WEB-INF目錄內。接下來部署該服務，因此需要deploy.wsdd 文件，見附錄D，如<圖6>



<圖6>deploy.wsdd

若要將服務移除的話，則需要undeploy.wsdd 文件，部屬的方式非常簡單，在終端機下，分別輸入以下指令指令:

```
[root@localhost ~]#cd /opt/tomcat/webapps/axis/WEB-INF
[root@localhost WEB-INF]#java -cp $AXISCLASSPATH
org.apache.axis.client.AdminClient
-lhttp://172.16.108.3:8080/axis/services/Multivariate
deploy.wsdd
```

成功時，便會出現以下的文字結果:

```
Processing file deploy.wsdd
<Admin>Done processing</Admin>
```

且在Apache Axis 主畫面中執行List 就會出現全部所有提供的服務名稱以及各個服務之下所提供的函數，若執行旁邊的WSDL，則會出現與<圖5>相同的WSDL 檔。

Step 5.撰寫顧客端程式

在撰寫顧客端程式的方法其實有很多種,在此論文中選用最基本最簡單的方法來撰寫，其程式碼請見附錄E。將MultivariatePCAClientTest.java編譯為Class檔，進入終端機輸入以下指令:

```
[root@localhost ~]#java  -cp $AXISCLASSPATH MultivariatePCAClientTest P_C_A 2 15 173,155,175,171,166,167,163,155,159,168,166,169,159,154,160,66,49,72,68,63,64,61,52,55,65,61,73,57,49,60
```

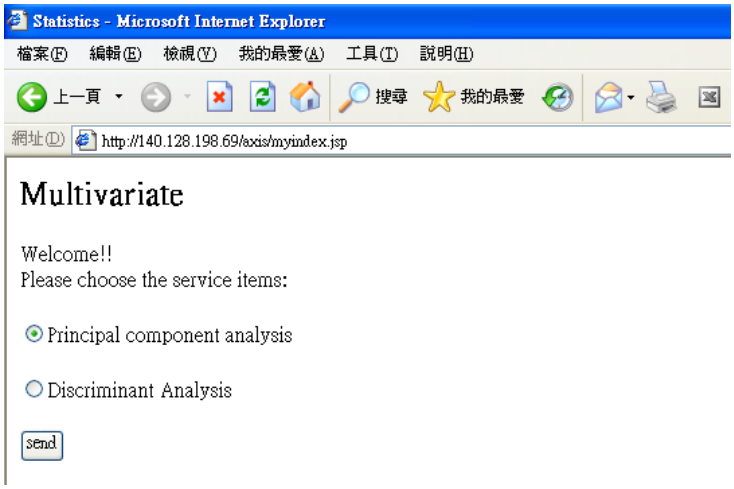
則執行結果會顯示出，th 0^{th} Component: 0.6658 0.7460 the 1^{th} Component: 0.7460 0.6658，可知這顧客端程式是可以運行的。

Step 6.撰寫JSP 網頁

在撰寫JSP動態網頁之前，先修改在/opt/tomcat/conf目錄下的server.xml檔案，並找到以下的程式碼:

```
<Connector port="8080"maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"  enableLookups="false"  redirectPort="8443"  acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true" />
```

將Connector port="8080"改為Connector port="80"，其目的是為了讓別台電腦能以網頁http://140.128.198.69/axis開啟。我們將已經撰寫完畢的myindex.jsp放置/opt/tomcat/webapps/axis目錄下，並且輸入http://localhost/axis/myindex.jsp 則會出現如<圖7>的畫面。



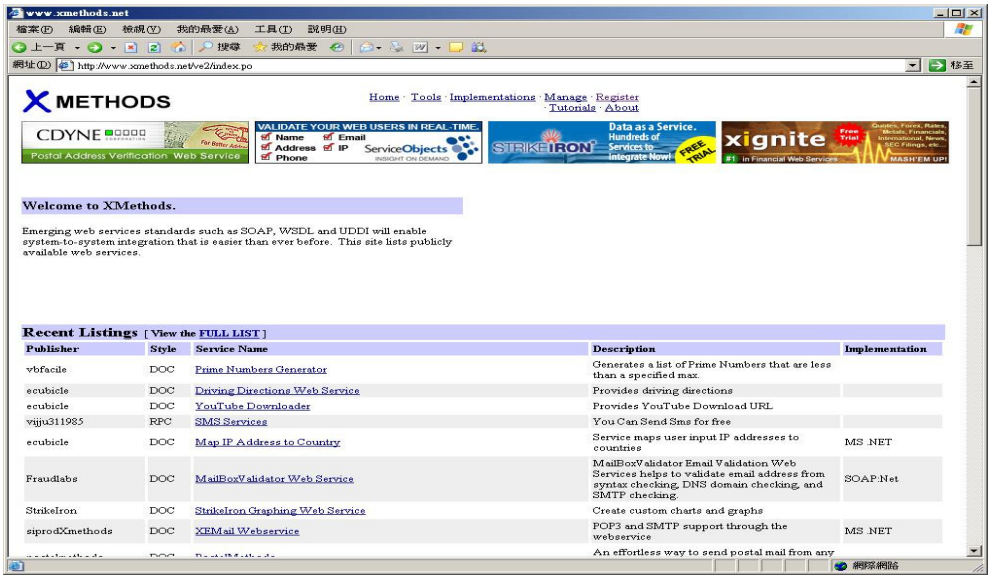
<圖 7>統計首頁

撰寫JSP網頁可參考呂文達(2006)的JSP 2.0動態網頁技術。首先建立首頁，再分別撰寫JSP網頁，將每個服務程式架構連接起來，其中包含了標題、資料輸入的方框、計算的按鈕等。當輸入資料後，點選按鈕後網頁所做的動作分為請求與回應兩部分，分別將兩部分用網頁形式呈現，請求部分為傳遞資料給伺服器的檔案內容，回應部分則為伺服器的傳遞給服務需求者的回傳值，兩者間的傳遞內容皆為XML格式。

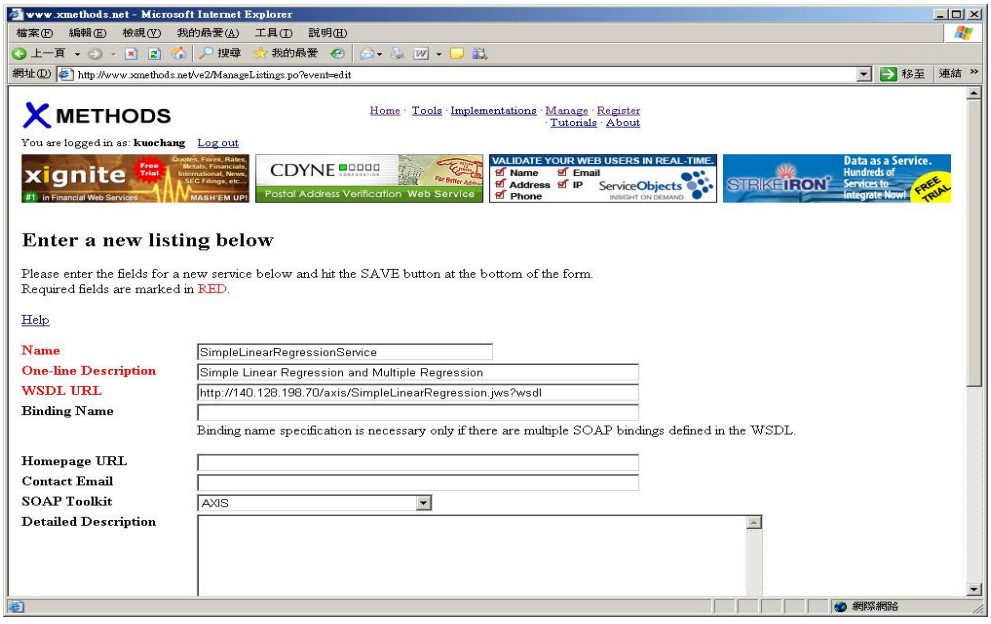
Step 7. 在Xmethods 網站註冊服務

在本篇文章中，我們選擇http://www.xmethods.net 這網站作為UDDI 服務仲介者的角色，<圖8>為其首頁，得需要經過註冊之後才可以在這網站發佈服務程式，註冊完畢後，執行Manage 裡的 Add a new listing，則出現<圖9>的畫面，將資訊填完畢之後執行Save

Service，Xmethods 需要一段時間的審查，才會將這服務正式的放入其網站裡。



<圖 8>XMETHODS 網站

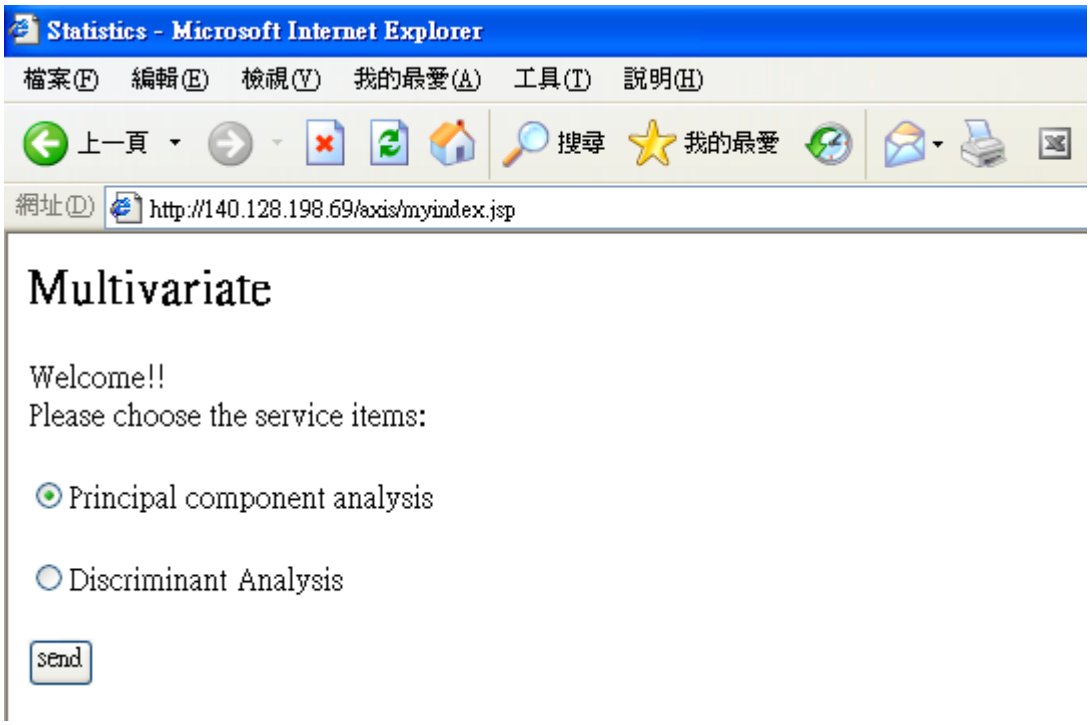


<圖 9>填寫服務資訊

Step 8. 實際測試網路服務

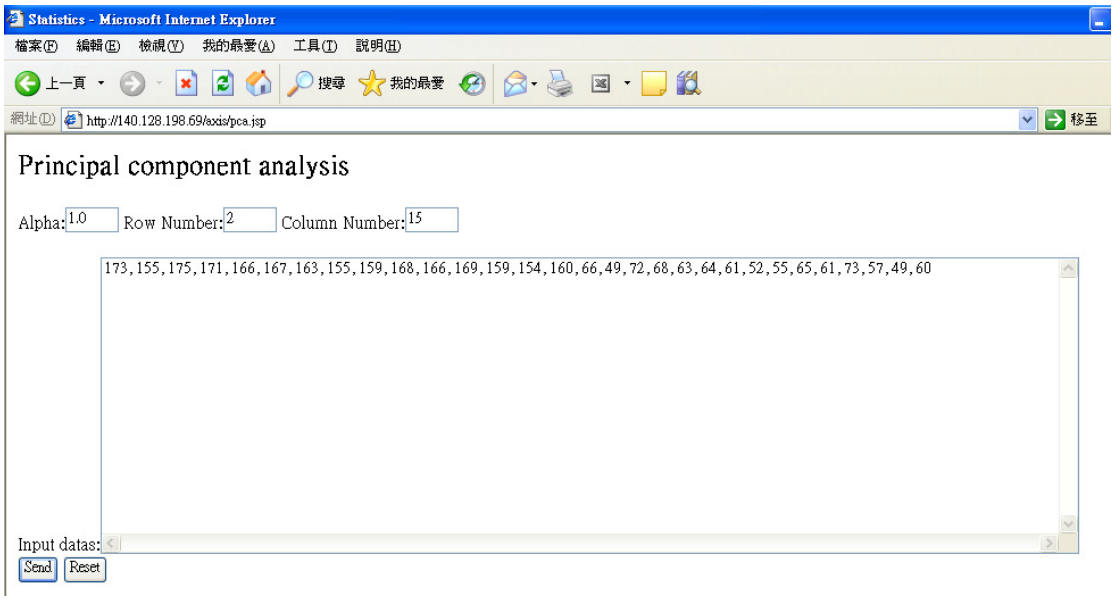
實際利用另一台電腦測試，輸入網址

http://140.128.198.70/web/index.jsp，進入統計計算首頁，如<圖 10>。



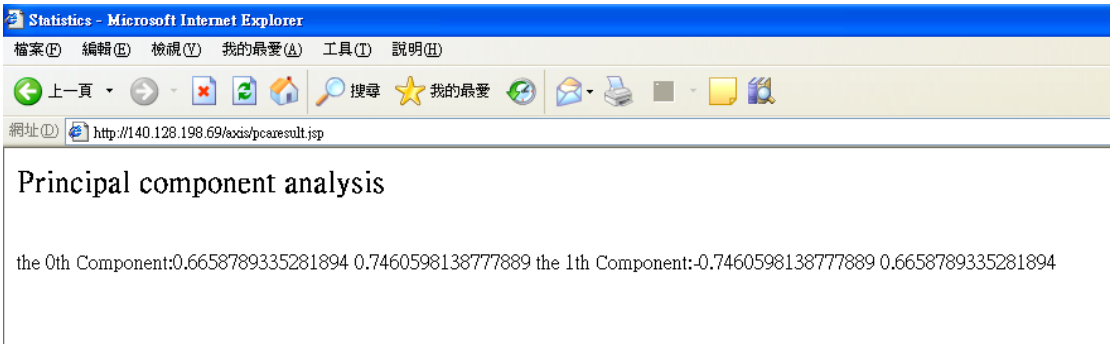
<圖10>統計計算首頁

執行Principal component analysis，出現Principal component analysis操作介面，如<圖11>。



<圖11>Principal component analysis操作介面

這當中要注意的是Input datas的欄位，依序將每一組變數的資料輸入，並以「，」隔開，輸入完畢按下send鈕便可以得到運算結果，如<圖12>。



<圖11>Principal component analysis運算結果介面

而有關Discriminant Analysis的部分，操作方式與Principal component analysis雷同，因此便不再多做贅述。

第伍章 結論

第一節、結論

本篇文章所談及之題庫系統，它結合了 XML 可擴展標記語言，透過 XML 的幫助，即便題庫設計者並不熟稔 JAVA 程式的撰寫，也可以很輕鬆的經由更改變數設定的方式，而製作出專屬的題庫系統，除了測驗之外，還可以運用在教學上。題庫系統提供了一套快速、立即、簡單的介面，不但替製作者省去了不少麻煩，也讓測驗者在使用上相當便利。

學習統計的人，都曾面臨過一個很大的問題，那就是為了得到結果，不得不去面對龐大數據的統計計算，如此大量的數據，不但運算起來費時，對於應用統計的人來說，也是一項相當枯燥的工作，因此若能借助軟體的幫助，必定能達到事半功倍的成效。很可惜的是統計軟體的費用都相當的昂貴，雖然近年來開始有免費的統計軟體誕生，但想使用這些軟體前，得先花一段時間去學習該軟體的相關語法，而每一套軟體都有屬於該軟體的語法，因此使用起來也相當的不方便。

本篇文章的統計計算服務讓初學者可以利用此網站，免費使用統計計算的服務，不需安裝任何軟體即可立即使用，特別的是，使用者不需要具備撰寫程式的能力，也可以使用統計計算的服務。而對於架設者，因為是使用 Linux 作業系統，它具備有開放程式碼的特性以及

無版權的性質，對於有特殊需求以及成本考量的人來說，無疑是相當合適的開發環境，也因為是開放式的程式碼，對於系統的漏洞也可以立即修補，而不需為了官方的修補漏洞程式苦苦等候。

現今網路應用越來越發達，平板電腦和智慧型手機也如雨後春筍般漸漸被社會大眾接受，因此透過 Web Services 的幫助一定可以讓統計服務，更廣泛的應用，變得更加方便、快速、立即、簡單。

第二節、未來研究方向

此處題庫系統所提供的測驗題型是以單選題、複選題以及填空題的順序進行，而且當測驗者中途更換章節，則原先所作答的結果將不予以儲存，換句話說，若測驗者沒有完成整個章節的答題將不會知道標準答案以及結果。而且在填空題的答案判定上，仍舊有一些小瑕疵，例如在檢查與正確答案是否相同時，會因答案當中是否包含空白字元，而造成判定錯誤，這些都是將來可望改進的方向。未來甚至可以結合資料庫系統，將題庫測驗變得立即。

統計的應用絕對不止本篇文章所提到的而已，還有更多常用的一些統計分析，若能將這些應用慢慢的架設起服務，一定可以讓此系統更加的完備，除此之外，若能結合資料庫系統(例如:Mysql)也能讓系統更加的健全。

附錄 A　vi 的使用

基本上 vi 共分為三種模式，分別是「一般模式」、「編輯模式」、「指令列命令模式」。

一般模式：

在一般模式底下，可以使用『上下左右』按鍵來移動游標，也可以使用『刪除字元』來處理檔案內容，以及『複製、貼上』來處理文件資料。直得注意的是，雖然在一般模式可以有以上的動作，但是卻無法編輯文件的內容，編輯的部分則要仰賴底下的編輯模式。

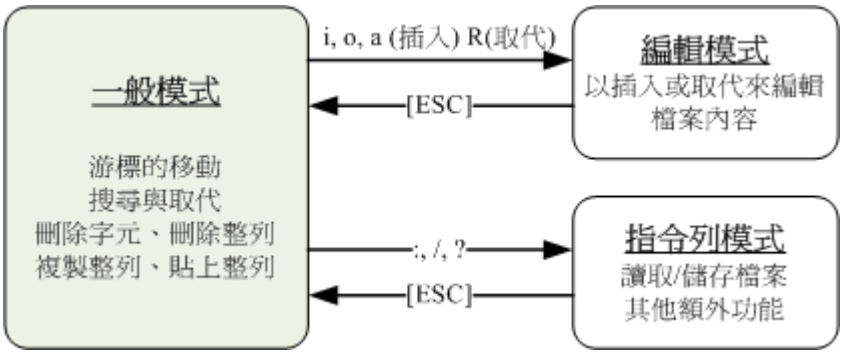
編輯模式：

在一般模式底下，當你按下『i、I、o、O、a、A、r、R』等任何一個字母才會進入到編輯模式，此時在畫面的左下方會出現『INSERT 或 REPLACE』的字樣，此時才可進行編輯，而在編輯模式下按下『Esc』按鍵時，便可退出編輯模式回到一般模式。

指令列命令模式：

在一般模式底下，輸入『:/?』三個當中的其中一個，便可將游標移動到最底下。此模式可以提供『搜尋資料』的動作，而『讀取、存檔、取代、離開、顯示行號』等動作，皆是在此模式底下完成，其中『儲存後離開』的指令為輸入『:wq』，而『離開不儲存』則為『:q』。

簡單的說，我們可以將三個模式以底下的圖來表示：



值得注意的是，一般模式可與編輯模式或指令列模式切換，但編輯模式與指令列模式並不能互相切換，也就是說，不論要切換到編輯模式或指令列模式，都必須先回到一般模式底下才有辦法執行。

附錄 B Tomcat 及 Axis 環境變數設定

在系統變數中新增 AXIS_HOME 變數，且變數值為/opt/axis。並且新增加 AXISCLASSPATH 變數，環境變數可以告知 Java 應用程式要到哪邊尋找第三元件，所謂第三元件也就是本身不包含在 JDK 裡的元件，而是另外擴充的元件。以下為完整的操作內容。

在終端機輸入以下指令:

```
[root@localhost ~]#vi /etc/profile
```

並在文章最下端加入下列文字:

```
export JAVA_HOME=/usr/local/java
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME:$JAVA_HOME/jre/lib/ext
export CATALINA_HOME=/opt/tomcat
export CATALINA_BASE=/opt/tomcat
export PATH=$CATALINA_HOME/bin:$PATH

CLASSPATH=.:$CATALINA_HOME/common/lib:$CATALINA_HOME/common/lib/servlet-api.jar:$CATALINA_HOME/common/lib/activation.jar:$CATALINA_HOME/common/lib/mail.jar:$CATALINA_HOME/common/lib/xmlsec-1.4.2.jar:$CATALINA_HOME/common/lib/javastatsoft_beta1.3.jar:$CATALINA_HOME/common/lib/xercesImpl.jar:$CATALINA_HOME/common/lib/xml-apis.jar

export CLASSPATH
export AXIS_HOME=/opt/axis
export AXIS_LIB=/opt/axis/lib

AXISCLASSPATH=.:$AXIS_LIB/axis.jar:$AXIS_LIB/axis-ant.jar:$AXIS_LIB/commons-discovery-0.2.jar:$AXIS_LIB/commons-logging-1.0.4.jar:$AXIS_LIB/jaxrpc.jar:$AXIS_LIB/log4j-1.2.8.jar:$AXIS_LIB/saaj.jar:$AXIS_LIB/wsdl4j-1.5.1.jar:$AXIS_LIB/mail.jar:$AXIS_LIB/xmlsec-1.4.2.jar:$AXIS_LIB/xml-apis.jar:$AXIS_LIB/xercesImpl.jar

export AXISCLASSPATH
export PATH=$AXIS_HOME:$AXIS_LIB:$PATH
```

附錄 C Multivariate.java

```
import javastat.multivariate.*;
import javastat.multivariate.PCA.*;
import javastat.multivariate.DiscriminantAnalysis;

public class Multivariate {
    public String P_C_A(String alpha,String rownum,String colnum,String
testscore) {
        double Alpha=Double.parseDouble(alpha);
        String[] testscores=testscore.split(",");
        int Rownum=Integer.parseInt(rownum);
        int Colnum=Integer.parseInt(colnum);
        double[][] Testscores=new double[Rownum][Colnum];
        int index=0;
        for(int i=0;i<Rownum;i++){
            for(int j=0;j<Colnum;j++){

    Testscores[i][j]=Double.parseDouble(testscores[index]);
                index++;
            }
        }
        String covariance="covariance";
        PCA pca = new PCA(Alpha, covariance,Testscores);
        int len=pca.principalComponents.length;
        String msg="";
        for(int i=0;i<len;i++){
            msg+=" the "+i+"th Component:";
            for(int j=0;j<pca.principalComponents[i].length;j++){
                msg+=""+pca.principalComponents[i][j]+" ";
            }
        }
        return msg;
    }
    public String Discriminant(String rownum,String colnum,String
testgroup,String testdata){
        int Rownum=Integer.parseInt(rownum);
        int Colnum=Integer.parseInt(colnum);
```

```
        String[] testgroups=testgroup.split(",");
        String[] testdatas=testdata.split(",");
        double[][] Testdatas=new double[Rownum][Colnum];
        double[] Testgroups=new double[Colnum];
        int index1=0;
        for(int i=0;i<Rownum;i++){
            for(int j=0;j<Colnum;j++){

    Testdatas[i][j]=Double.parseDouble(testdatas[index1]);
                index1++;
            }
        }
        int index2=0;
        for(int i=0;i<Colnum;i++){
            Testgroups[i]=Double.parseDouble(testgroups[index2]);
            index2++;
        }
        DiscriminantAnalysis da=new
 DiscriminantAnalysis(Testgroups,Testdatas);
        int len=da.linearDiscriminants.length;
        String msg="";
        for(int i=0;i<len;i++){
            msg+=" the "+i+"th Discriminants:";
            for(int j=0;j<da.linearDiscriminants[i].length;j++){
                msg+=""+da.linearDiscriminants[i][j]+" ";
            }
        }
        return msg;
    }
}
```

附錄 D deploy.wsdd 與 undeploy.wsdd 文件

* deploy.wsdd

```
<deployment name="test" xmlns="http://xml.apache.org/axis/wsdd/"
    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
    <service name="Multivariate" provider="java:RPC">
       <parameter name="className" value="Multivariate"/>
       <parameter name="allowedMethods" value="*"/>
    </service>
</deployment>
```

* undeploy.wsdd

```
<undeployment name="test" xmlns="http://xml.apache.org/axis/wsdd/">
<service name="Multivariate"/>
</undeployment>
```

附錄 E　顧客端程式

```
import java.io.PrintStream;
import java.net.URL;
import javax.xml.rpc.ParameterMode;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.utils.Options;
public class MultivariateClientTest{
    public MultivariatePCAClientTest(){
    }
    public static void main(String args[])throws Exception
    {
        Options options = new Options(args);
        String endpoint =
"http://localhost:8080/axis/Multivariate.jws";
//       String endpoint =
//"http://172.16.108.3/axis/Multivariate.jws";
        args = options.getRemainingArgs();
        if(args == null || args.length !=5 )
        {
            System.err.println("worng datas input");
            return;
        }
        String method = args[0];
        if(!method.equals("P_C_A")
        && !method.equals("Discriminant")
        ){
            System.err.println("wrong method");
            return;
        } else
        {
            String i1=args[1];
            String i2=args[2];
            String i3=args[3];
```

```
            String i4=args[4];
            Service service = new Service();
            Call call = (Call)service.createCall();
            call.setTargetEndpointAddress(new URL(endpoint));
            call.setOperationName(method);
            String ret = (String)call.invoke(new Object[] {i1,
i2,i3,i4});
            System.out.println(ret);
            return;
        }
    }
}
```

附上關於主成分分析服務的顧客端程式，另外區別分析服務的顧客端程式與這個程式非常的雷同，只是所在的URI 位置以及使用的方法、參數不相同，因此就不附上程式碼。

參考文獻

01. 陳順宇(2000)。多變量分析。台北：華泰書局。
02. 賴俊安、林俊良（2002)。Java2 設計實務-徹底研究。美商麥格羅•希爾國際股份有限公司。
03. 蔡德明(2007)。鳥哥的 Linux 私房菜。台北：上奇科技。
04. 何家益、黃世陽、李篤易、張賀閔（2007)。JAVA 2 程式設計從零開始。台北：金禾資訊。
05. 呂文達（2006)。JSP 2.0 動態網頁技術。台北：博碩文化。
06. 蔡德明(2003)。鳥哥的 Linux 私房菜 伺服器架設篇。台北：上奇科技。
07. 林上傑、林康司（2004)。JSP2.0 技術手冊。台北：碁峰資訊。
08. 黃中杰（2002)。JAVA 與 XML 技術手冊。台北：碁峰資訊。
09. 艾克斯坦（2002)。Java Swing ， 基礎篇。台北：歐萊禮。
10. 羅依（2003)。Java Swing ， 進階篇。台北：歐萊禮。
11. 吳國彰（2009)。網路服務在統計與繪圖上的分析。台中：東海大學。
12. 王詩涵（2008)。Web Services 在統計上的應用。台中：東海大學。
13. 昇陽網站(http://java.sun.com)。
14. AXIS 網站(http://ws.apache.org/axis)。