Using SAS for analyzing Website traffic

Chong Ho Yu, Tempe, AZ
Angel Jannasch-Pennell, Tempe, AZ
Samuel DiGangi, Tempe, AZ
James Carvalho, Tempe, AZ

Abstract

Tracking Web traffic is crucial to both corporate and educational institutions. Analysis of user patterns could provide instructive feedback for continued improvement of websites. This paper demonstrates how SAS/CORE (SAS Institute, 1990a), SAS/GRAPH (SAS Institute, 1990b) and JMP (2000a) can be programmed to extract useful information from website user access log.

Introduction

Analysis of user access log of a web server is helpful in many ways. For instance, an instructor who offers web-based courses can find out which students access the site more often, what pages are more popular, when the prime time is, and so on. This information can provide an empirical base for the instructor to modify instructional strategy. However, the raw access log is only a text file and thus is difficult to read. The following is an example of user access log data:

```
  134.139.129.33 - - [04/Mar/1998:17:05:32 -0700] "GET
/~alex/home.html HTTP/1.0" 200 3562
  134.139.129.33 - - [04/Mar/1998:17:06:18 -0700] "GET
/~alex/home.GIF HTTP/1.0" 200 4608
  204.38.7.171 - - [04/Mar/1998:17:06:50 -0700] "POST
/alex/search.html HTTP/1.0" 200 15458
  204.38.7.171 - - [04/Mar/1998:17:06:50 -0700] "GET
/alex/button.gif HTTP/1.0" 200 5856
```

Today there are quite a few software tools for analyzing a user log. Some of them such as AccessWatch (Maher, 2000) are CGI-based and thus it is difficult to customize them. On the other hand, some such as LinkMinds (Macromedia, 2000) are more flexible but are extremely expensive. The advantages of using SAS products to analyze user logs are: (1) a SAS programmer can manipulate the data in the way he/she wants, and (2) SAS products are available in many institutions. This paper will illustrate how you can clean up the access log and present useful results for evaluation using SAS products.

SAS/CORE and SAS/GRAPH

Linking user log with SAS

If SAS/CORE is not installed on the same server where the Web server is situated, one must create a link between SAS and the targeted user log. There are several ways to accomplish this task. One can mount a Network File System (NFS) volume that can be recognized by both computers, or you can simply issue a "FTP" command in SAS (see below). After the link is established, SAS can read the data in the data step.

```
filename indata FTP 'filename_of_the_access_log'
cd='full_path_of_the_access_log'
user='your_login_name' host='web_server_hostname'
pass='your_password';
```

Extracting relevant data

Not everything in the user log is useful. Typically what can be used are the IP address (From where did the users look at your website?), the date and time (When did they read it?), and the hit (What did they look at?) The programmer can use dummy variables as shown below to skip all unnecessary data. To save memory space, one should drop all dummy and temporary variables after the data step is finished.

```
Data one (drop=temp2 temp dummy 0-dummy7 datetime);
        infile indata firstobs=3 missover;
length ip $ 15 datetime $ 21 hit $ 40 temp2 $ 40
    date $ 2 mth $ 3 year $ 4 hour $ 2 min $ 2;
input ip $ dummy0 $ dummy1 $ datetime $ dummy2 $
    dummy3 $ page $ dummy4 $ dummy5 $ dummy6 $
    dummy7 $;
```

The above selection works very well if the primary function of the website is to display information. However, one may want to read more data if the server performs functions other than displaying pages. For example, in the previous user log example, right after the date and time data there are "action" data, which contain either "get" or "post." If a user reads the displayed information only, the action is "get." If the homesite allows a user to submit a query to search a database or upload a form to your server, then the action will be "post." A web-based instructor may want to find out the ratio between "get" and "post." (How many students use the search engine? How many students post questions or submit homework?)

Usually the user log may contain headers or some blank space in the beginning, therefore one should check the user log first to find out which line should be the beginning of the actual data. In this example, line three is the beginning and thus "firstobs=3" is inserted in the source code. Further, the user log may have missing data at the end of the line. To avoid misreading the data, the code "missover" must be included.

Some variables have fixed-length values such as "IP Address," but some have varied-length values such as

"hit." (e.g. both strings "./alex/computer/sas/sas.html" and "./alex/computer/spss/ver10/group4/lesson1/example/example10.jpeg" are "hits" but they vary in length) Variables with varied-length string could cause misreading of data. Therefore one should assign a proper length to each variable.

To extract user activity data of a particular website from the user log, one should retain only the relevant data and delete all others. One can check which entries are related to the target website by using a **substring (substr)** function. In this example (see below), the **substring** function starts to scan the "hit" from the second position and check four columns only. If the entry is "/alex/index.html", the function would keep the record, otherwise it would erase the record in the memory of the SAS program (not in the actual user log).

```
if substr(page,2,4) ne 'alex' then delete;
```

Cleaning up IP numbers

When a Webmaster is developing a website, he may check the site very often. These counts are misleading for analyzing the website traffic. Therefore, it is recommended to delete all entries that originated from the workers who are involved in developing the pages. To do this, simply discard the records if the IP numbers belong to the workers (see below).

Moreover, it is helpful to find out how many accesses occur within the organization and how many are from external web surfers. The first sixteen bits of the IP address indicate the origin of the access. One can use the **substring** function from SAS to classify the IP address into two groups: "Campus-access" and "Outside-access".

```
if ip in ( "129.219.12.22", "129.219.11.46",
"129.219.19.14","129.219.48.15",
"129.219.12.30","129.219.11.17")
   then delete;
if substr(IP,1,7) = "129.219" then IPAdd = "Campus-
access";
   else IPAdd = "Outside-access";
```

Cleaning up page access

In addition, it is important to distinguish "hit" from "page access." A hit includes every object accessed by the user such as individual JPEG images, GIF images, WAV sound clips, HTML files and so on. If a user opens a Webpage, which has two JPEG images and three GIF images, the total number of hits would be six. This statistic artificially inflates the website traffic. On the other hand, a page access counts only the html page. It is recommended using the later because it reflects the usage of the website in a more accurate manner. The following SAS code could perform the filtration task:

```
temp = reverse(scan(reverse(hit),1,'.'));
if upcase(temp) in ('GIF','JPG','JPEG') then delete;
    page = hit;
    page=put(page,$char40.);
```

To exclude non-HTML hits, use the **reverse** function to reverse the "hit" string. Then use the **scan** function to locate the extension (.html, .jpg, .gif...etc). Because the extension names may be in both upper and low cases, use the **upcase** function to convert them into capital letters. Next, use an if-then statement to delete all non-HTML hits. In this example, only JPEG and GIF images are taken out. In your own implementation, you can take out other types of hits such as Shockwave movies, Java applets, QuickTime movies, and Wave sound clips. After the hits are cleaned up, assign the values of hit into a new variable named "page," which stands for "page access."

Hotlinking page access

If the result is displayed via SAS/InteNet, the variable "page" can be used to show the frequency of page access of each webpage. One can hotlink the display of the pages by using the following method:

```
start="<a href=http://seamonkey.ed.asu.edu";
middle=">";
end="</a>";
link=trim(page)||middle||trim(page)||end;
link2=start||link;
```

To make a hotlinked text, first create several constants. The constant "start" carries the string of the website and the HTML tag of starting a link. The constants "middle" and "end" contain the tags that bracket the text to be hotlinked. To show a hotlinked text on the Web, concatenate the preceding constants and the variable "page" in the proper order. To avoid empty tails, use the **trim** function to remove the tail of shorter string.

Cleaning up date and time

Date and time in the user log are together in a continuous string. The code below divides the variable "datetime" into several different variables. First, the **substr** function extracts date and time. Second, the **compress** function removes the slash so that the date/time string conforms to the standard SAS date/time format. Next, use different date functions to extract the month, year, hour, minute, and date information from the string.

```
dt=input(compress(substr(datetime,2,17),'/'),datetime15.);
  month=month(dt);
  year=year(dt);
  hour=hour(dt);
  min=minute(dt);
  date=day(dt);
```

Erasing duplicates

When the network connection is slow, the user may click the refresh/reload button several times within a minute. This action improperly inflates the number of page accesses. To avoid duplication, one can perform a sort of **no duplicated unique key (nodupkey)** on the variables "IP," "dt" and "page" (see below). If the same person looked at the same page at the same time (within one minute), only the first page access will be kept.

```
proc sort nodupkey;
   by IP dt page;
```

Counting page access

Now the access log is ready for analysis. In the following, **proc summary** is used to display the frequency count of each page access. Also, the displayed pages are hotlinked.

```
data two (drop=temp); set one;
   count = 1;
proc summary data=two;
   class page; var count; output out=new sum= ;
   proc sort; by descending count;
   proc print noobs data=new; var link2 count;
   title "Page access ranked by the number of accesses";
```

Counting page access by month

In the following, **proc summary** and **proc chart** are used to show the page access by month:

```
proc sort data=two; by month;
proc chart data=two;
   hbar month /group=year midpoints=1 2 3 4 5 6 7 8 9
10 11 12;
   title "Page access by month";
proc summary data=two;
   class month; var count; output out=new sum= ;
proc sort; by descending count;
proc print noobs data=new; var month count;
   title "Page access ranked by months";
```

Counting page access by hour

The following module returns the page access ranked by hour:

```
proc chart data=two;
   hbar hour;
   title "Page access by peak hour";
proc summary data=two;
   class hour;  var count; output out=new sum= ;
proc sort; by descending count;
proc print noobs data=new; var hour count;
   title "Page access ranked by hour";
```

Counting page access by location

The following code segment shows the page access by location ("Campus-access" or "Outside-access"):

```
proc chart data=two;
   hbar IPAdd;
   title "Page access by location";
proc summary data=two;
   class ipadd;  var count; output out=new sum= ;
proc sort; by descending count;
proc print noobs data=new; var ipadd count;
   title "Page access ranked by location";
   run;
```

Now the programmer can convert the SAS output in Web form by using SAS/IntrNet. Illustration of using SAS/IntrNet is beyond the scope of this paper. Interested readers may consult SAS Institute (2000b).

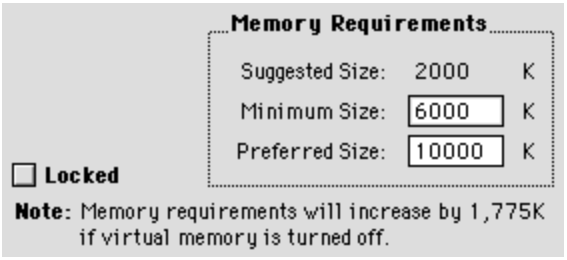
JMP

JMP is a GUI-based exploratory data analysis package, which is relatively user-friendly. The trade-off of this user-friendliness is the lack of powerful statistical functions and programming flexibility. Further, JMP is not web-enabled and thus all analysis must be performed within one computer. Nevertheless, JMP is sufficient for basic web traffic analysis. JMP has both Mac and PC versions. This illustration is based upon the Mac version.

Allocate memory

Since a web log is usually huge, the default allocated memory of JMP may not be able to handle a file with several thousand entries. Therefore, one should increase the memory size of JMP in the following procedure:

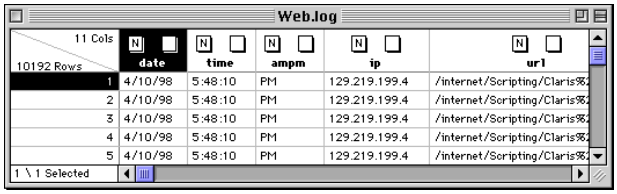
- Make sure JMP is not open. Memory cannot be allocated when the application is active.
- Highlight the icon of JMP.
- Select **Get Info** from **File** simultaneously.
- A pop-up window will show you the system info of the application. In the bottom box entitled **Memory Requirements**, enter 10000 for **Preferred size** and 6000 for **Minimum size**. Then close the pop-up window.



Clean import

In this example both Web user log and JMP are located in the same hard drive, and therefore JMP can

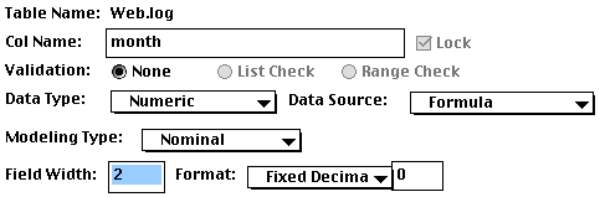
open the log directly. One nice thing about JMP is that the log can be imported without any cleaning effort, and surprisingly, the table is as clean as the following. Then one can name the fields accordingly. Once the field is correctly defined and the file is saved, one may always import the user log data properly.



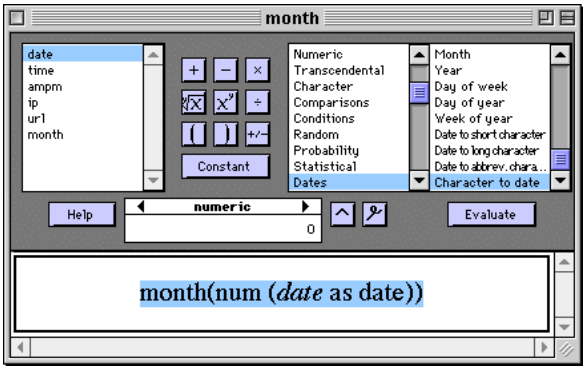
Extract month and year

The imported table is not good enough for analysis, of course. For example, date, month, and year are all in one field, but time and am/pm are separate. Follow the procedures below to extract month from date:

- Double click a blank column to create a new field.
- Name the new field as **Month**.
- In the dialog box, select **numeric** as **data type** or leave it unchanged. If the selected data type does not match the actual data nature, JMP will adjust it for you.
- Define the **field width** as 2.
- Select **Formula** for data source and click OK.



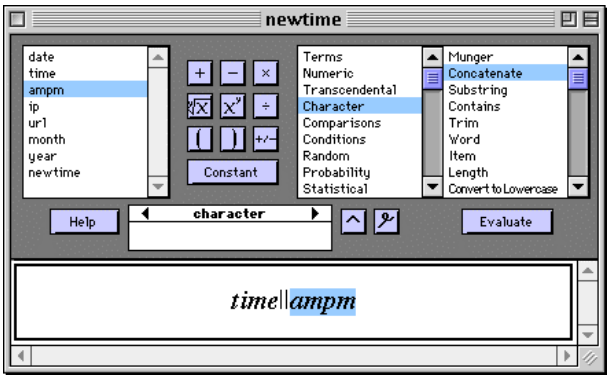
- In the formula box, select the function group **Dates**.
- Select the function **month**.
- Select the variable **date**.
- Select the function **Character to date**.
- Close the box and the month will be extracted into the new field you just created.
- Use the same method to extract **year** from **date**.



Concatenate time and am/pm

Follow the following procedures to concatenate time and am/pm:

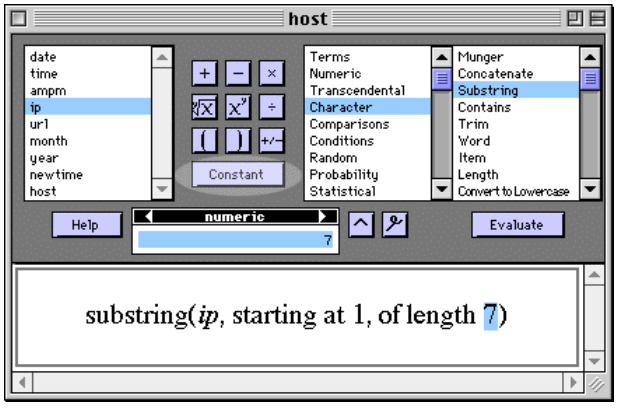
- Create a new field named **newtime** and select the proper **data type**, **data source**, and **field width**.
- In the formula box, select the function **Concatenate** from **Character**.
- Select the variables **time** and **ampm**, then close the box.



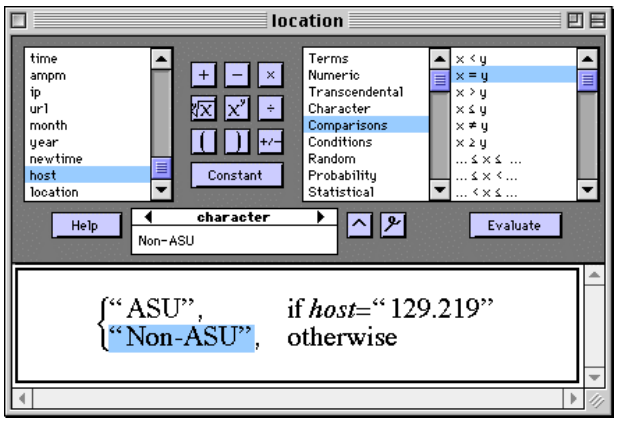
Extract network addresses

One may want to know how many hits are from internal users and how many are from outside. In this example, it is assumed that the Website is situated in a university. Universities use Class B addresses and therefore the first two chunks of the IP addresses (129.219) denote the network addresses and the last two show node addresses. In other words, one can classify the sources of hits by looking at the first two portions of IP numbers. The following procedures are used to extract network addresses from IP numbers:

- Create a new field labeled **host** and select the proper **data type**, **data source**, and **field width**.
- In the formula box, select the function **Substring** from the function group **Character**.
- Select the variable **ip** and put "7" in length.
- Click on the button **constant** so that "7" is registered, then click OK.



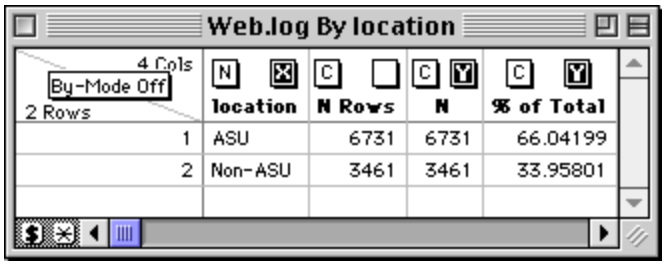
- Create a new field labeled **location** and select the proper **data type, data source**, and **field width**.
- In the formula box, choose the function **If** from **Conditions**.
- In the result box for the first condition at the left, type ASU and click **constant**.
- For the if statement select the variable **host**.
- Choose **x=y** from the function group **Comparison**.
- Enter **129.219** as the first condition.
- In the result box, for all other conditions (otherwise) type **Non-ASU** and click constant.
- Close the box.



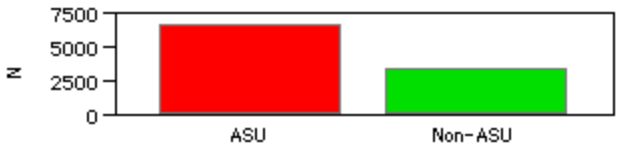
Summary of location by month

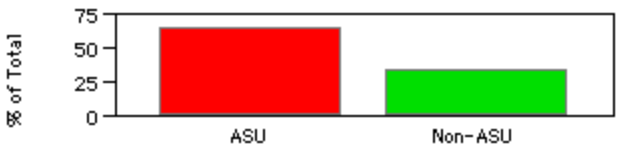
Now it is ready to conduct analysis. Let's start with something simple. To know the number of hits by location (ASU/Non-ASU), the following task should be performed:

- Choose **Group/Summary** from the pull down menu Tables.
- Select **location** as **Group**.
- Select **N** and **% of Total** from the pull down menu **Stats**.
- Click OK. A summary table is presented as the following.

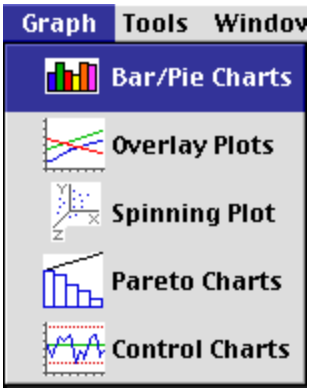


- To see the info in graphics, select **Bar/Pie Charts** from **Graphs**.

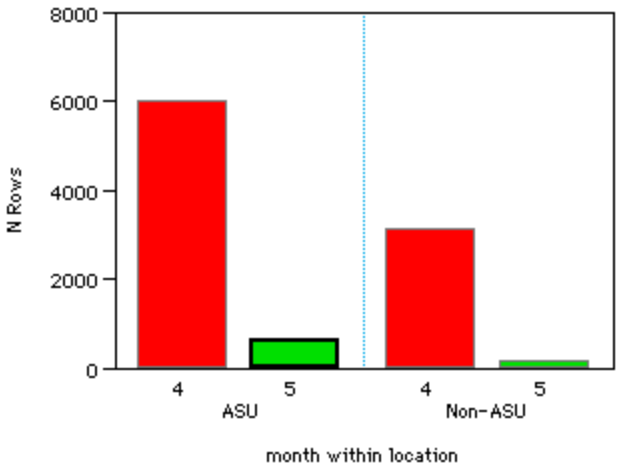




- To see the hits by location with a break down by month, go back to **Tables**, in addition to location, put **month** into **Group**.
- When a summary table is available, select **Bar/Pie Charts** from **Graphs** again.



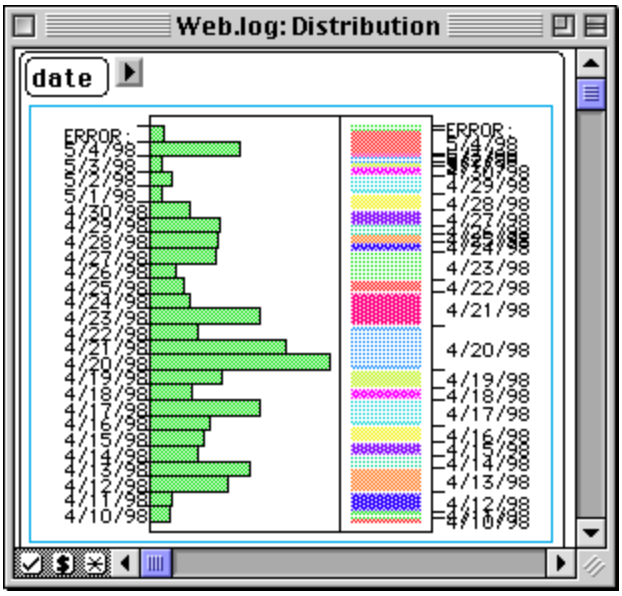
- In the dialog box put **N** Rows in **Y**. It will create bar charts with month as the break down.



Daily accesses

To check the number of hits by day, use the following procedures:

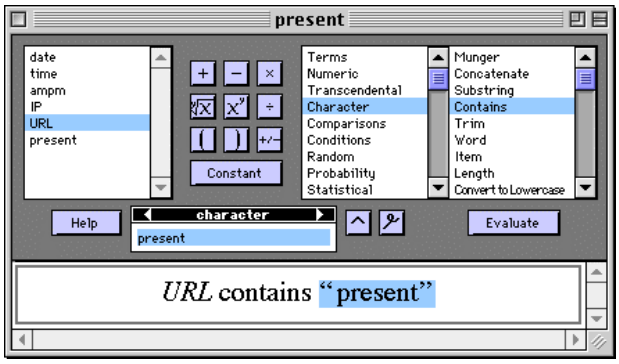
- Choose **Distribution of Y** from **Analyze**.
- Select **date** from the variable list
- Click on the button **Add**
- Click OK



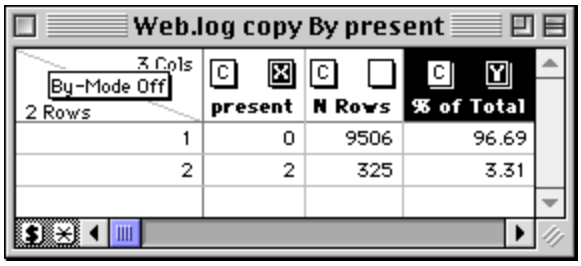
Summary of accessed section

One may be interested in knowing the number of page access of a particular section. For example, if you have a folder name "present," which contains your presentations, you can find out the number of page accesses of "present" and its percentage of total by the following procedures:

- Create a new field called **Present**. Select **Formula**.
- In the formula box, choose **Contains** from **Character**.
- In the formula choose **URL** as the variable to be evaluated.
- In the formula, type present as character and press the button **constant**. The function will check whether the variable URL contains the string "present." If it is true, it will return the value "2," otherwise, "0."



- Choose **Group/Summary** from **Tables**.
- Choose **Present** as **Group**.
- From **Stats** choose **% of Total**.
- A summary table will be presented. Double click the field **% of Total** and select **Fixed Decimal**. Enter 2 for decimal positions.



The above summary table tells you that there are 325 page accesses in the section "present," which is 3.31% of the total.

Conclusion

No doubt tracking Website usage is crucial to Website improvement. This paper shows that rather than installing expensive user log analysis software package, one can use available SAS products to perform this function. One of the major advantages of using SAS products is that there are many statistical procedures that can be run to gain a deeper insight of the user pattern. To achieve this goal, SAS/STAT is required in addition to SAS/CORE and SAS/GRAPH.

Acknowledgement

Special thanks to Mr. Eldon Norton for improving the efficiency of the SAS codes in this paper. Also, many thanks to Dr. Barbara Ohlund for reviewing this paper.

References

Macromedia, Inc. (2000). LinkMinds. [On-line] Available URL: http://www.macromedia.com/ebusiness/

Maher, D. (2000). Accesswatch. [On-line] Available URL: http://www.accesswatch.com

SAS Institute. (2000a). JMP Start Statistics Version 4.0 for Windows. Cary, NC: the Author.

SAS Institute. (2000b). SAS/IntrNet and Web Publishing Tools Documentation. [On-line] Available URL: http://www.sas.com

SAS Institute. (1990). SAS procedures guide. Cary, NC: the Author.

SAS Institute. (1990). SAS/GRAPH guide. Cary, NC: the Author.