Paper 100-2012

Analysis of Clickstream Data Using SAS®

Sumit Sukhwani, Satish Garla and Goutam Chakraborty,
Oklahoma State University, Stillwater, OK

ABSTRACT

Analyzing Web data has become a must-have for businesses. Significant research has been done in studying clickstream data to understand the navigation behavior of users after visiting a Web site. Analyzing clickstream data is not easy for most companies because Web logs are stored in a form that is not suited for analysis. Before any meaningful analysis can be done, much effort is spent in transforming server logs to the right form so that they can be analyzed. This is one of the reasons why companies often use third-party services (such as Webtrends, Adobe, or Google Analytics) to analyze their Web log data. This paper demonstrates applying SAS macro programming to prepare a SAS data set from raw Web logs and to generate summary reports.

INTRODUCTION

Advancement in technology and growing use of the internet has opened up different study areas for statisticians. Every time users visit websites; clicks are saved that can be used for extracting useful patterns [2]. Clickstream data could be considered as a very rich source of information, because they contain behavioral information of the web site visitor. However it is difficult to analyze since it is available as unstructured data [3] and many different formats depending on the web server. Many companies have their specific ways of collecting and analyzing data; for example, e-commerce companies can measure the sales and demand of their products and identify behavioral patterns of consumers. Even non-profits such as universities are using their web data to market their courses [4].At times' clickstream data may be very difficult and costly to manage for e-commerce companies who would be using data for their businesses. Dealing with these challenges has compelled companies to purchase web analytical tools [6]. These tools range from simple reporting applications to much advanced analytical software applications like Google Analytics. SAS Web Analytics tool is among the popular sophisticated tools which help companies in analyzing and visualizing their web log data.

Most of the web analytics tools directly take web logs and give end users information in the form of charts, plots and reports. The end user lacks control over the raw data which if they had in an useful format(such as SAS data set) can be used for various other types of analysis which are not available in the tool. For example, one can get excellent insights by using Google Analytics for your website. However, businesses cannot perform advanced analytical methods like sequence analysis or social network analysis because they do not have the data in the right form. Nowadays companies have started integrating customer-level behavior data from a website into their analytics environment. In such cases it is important that companies have control over their web log data and make it available in the right form for other enterprise applications to use it. The macro discussed in this paper provides a user with a SAS dataset of weblogs processed in a form that can be used easily for any type of statistical analysis or modeling.

DATA COMPONENTS

SERVER WEBLOGS

Weblog can be defined as an electronic record of internet usage collected by web servers. Each web server has a separate configuration and settings which sometimes distinguishes weblog information from one server to another. The W3C maintains a standard format for web server log files, but other proprietary formats exist. Each record in the log usually contains IP address, html page name, date and time, referrer and additional information based on how it is setup. But these are the main elements that one will find in any setting. These logs can be stored as single file or can be separated as access logs, error logs, distinct logs etc. Site administrators usually have complete control over these files. We used a weblog collection with 6,633 entries collected over a week's time from a website. The name of the website is masked for confidentiality reasons.
The information contained in the web log for each user includes following items.

Visitor Identification Number: This is a unique identification number for each user visit. In the case of this client company, the server was configured to create two separate variables that capture the unique identification number.

Date and time: Timestamp of the page visit.

IP Address: Every machine has a unique address. This field captures the IP address of the machine from where the page request is originating.

Page URL: URL of the current page the user is viewing.

Referral Page Information: Referral page captures the URL of the source page from where the request has originated

Browser and device information: Browser and device column provides information on type of browser and device used for accessing the web pages. Earlier we have just seen these requests coming from desktops or laptops. Now we find various mobile devices like smart phones and tablets that are used for accessing web pages.

Sample weblog

```
41521390 2011-01-01 00:25:42 2.111.94.18 Mozilla/5.0 (Macintosh; U; Intel Mac
OS X 10_6_5; en-us) AppleWebKit/533.19.4 (KHTML, like Gecko) Version/5.0.3
Safari/533.19.4 "http://www.cokstate.edu/welcome/"
"https://www.google.com/#sclient=psy-
ab&hl=en&source=hp&q=oklahoma+state&pbx=1&oq”
```

Figure 1 Sample web log

Figure 1 shows a sample web log record. A SAS Data Step program can used to prepare a SAS Data set from this raw weblog. Table 1 shows the values in the SAS Data set after identifying variables for the elements in the weblog.

Variables	Information
Visitor Identification Number	41521390
Date and time of visit	2011-01-02 00:55:13
IP Address of the system	2.111.94.18
Page URL	"http://www.cokstate.edu/welcome/"
Referral Page Information	https://www.google.com/#sclient=psy-ab&hl=en&source=hp&q=oklahoma+state&pbx=1&oq”
Browser and device information	Mozilla/5.0 (iPad; U; CPU OS 4_2_1 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148 Safari/6533.18.5

Table 2 Categorization of weblog information

DATA PREPARATION

READING FROM WEBLOG

Weblogs can be extracted as .txt files from the server. If you are analyzing only access logs then all other types of logs like error logs should be filtered before you start making a SAS data set. This task can be easily performed by your system administrator. The macro discussed in this paper only works with access logs. The first step in the macro is to read and convert the .txt files to a SAS data set. The macro takes information from the log file and assigns appropriate SAS data types and formats. It is required you understand the structure of the data in your web log so that you can modify the macro to suit your server environment. The full macro code is reported in appendix. Figure 2 displays a sample of web log entries after these were converted into a SAS data set. As mentioned before, in the case of this client company, we had two variables (Visid_High and Visid_Low) representing the unique identification number. This may not be the case with other servers. Due to the differences in the way web logs are structured, you may have to tweak the macro slightly in this step to accommodate these differences.

Visid_High	Visid_Low	Date_Time	IP	Browser	Page_URL	Referral
41516042	3406734837	2011-01-01 00:2...	2.121.124.10	Mozilla/5.0 (iPad;...	http://thesprt.sportmania.com/thespr...	http://thesprt.sportmania.com/
41516042	3406734837	2011-01-01 00:2...	2.121.124.10	Mozilla/5.0 (iPad;...	http://thesprt.sportmania.com/thespr...	http://thesprt.sportmania.com/
41516042	3406734837	2011-01-01 00:2...	2.121.124.10	Mozilla/5.0 (iPad;...	http://thesprt.sportmania.com/thespr...	http://thesprt.sportmania.com/
41516042	3406734837	2011-01-01 00:2...	2.121.124.10	Mozilla/5.0 (iPad;...	http://www.sportmania.com/page/Sh...	http://thesprt.sportmania.com/cg...
204080579	470294790	2011-01-01 00:5...	12.42.5.195	Mozilla/5.0 (iPod;...	http://thesprt.sportmania.com/thespr...	http://thesprt.sportmania.com/m...
204080579	470294790	2011-01-01 00:5...	12.42.5.195	Mozilla/5.0 (iPod;...	http://thesprt.sportmania.com/thespr...	http://thesprt.sportmania.com/m...

Figure 3 Reading variables from weblog to form initial SAS dataset

You can also see from figure 2 that the macro identifies appropriate formats for the variables. Web logs in this stage are still in a form that cannot be used for data mining or web analytics. Each record in the data set represents a single page visit per user with the latest visit at the bottom. Each entry captures the time of visit for a page. In order to calculate the time spent on a page you should know the time of visit for the next visited page and this goes on for all other pages until the visitor exits the web site. Therefore, you can never calculate the time spent on the last visited page in any session.

CREATING OUTPUT DATS SET

Once the raw SAS data set is available, we can use SAS programming to transform the raw SAS data set into a form that can be used for analysis. The structure of the output data set can be formulated based on the type of analysis an analyst wants to perform. New variables need to be created in order to extract insights from the data. This can include creating simple variables such as “Browser Type” and “Date” to complex variables like “Session Duration” and “Percent Page Duration”. The macro developed and reported in this paper creates these new variables with processed information but also retains the raw variables from the input data set. The new variables that are created by this macro are explained below:

Time Spent: This variable captures the time spent by the visitor on each page. The time spent on the page can be calculated only by knowing the start time of the next visited web page which is available only in the next following observation. We used SORT procedures to reverse the order of data along with RETAIN statements to calculate the time spent on a page.

Session: A session is defined as a series of page requests from the same uniquely identified client with a time of no more than 30 minutes. We track the time spent information to calculate the session for a visit.

Session Duration: Session duration captures the total time spent on all the pages visited in a session.

Page Name: Page Name is the actual page visited by the user. The macro identifies this page as the name with .htm or .html extension as found in the complete URL. If there is not .an html or .htm page, the last string in the URL is taken as the page name. Table shows two different examples for page names.

URL	Page
http://www.athletics.okstate.edu/page/TV/LiveMatches/010268.html	010268.html
http://www.osu.okstate.edu/welcome	welcome

Table 2 Example of URL page and page name

Exit Page: This is the last page visited by user. This value is identified based on the session.

Percent of Pages Visit: This variable captures the number of times a page was visited in a particular session in percentage.

Percent of Page Duration: This variable captures the amount of time spent on a particular in a session in percentage.

Id	Page_Name	Time_Spent	Session	Session_Dur	Times_Visited	Pct_Page_Visit	Pct_Page_Duration
12770837772...	0,,10268,00.html	0.42	1	5.20	3	37.50	8.01
12770837772...	0,,10268,00.html	0.82	1	5.20	3	37.50	15.71
12770837772...	?target=http%3A%2F%...	0.53	1	5.20	1	12.50	10.26
12770837772...	0,,10268,00.html?target...	0.43	1	5.20	3	37.50	8.33
12770837772...	0,,10268,00.html?target...	0.38	1	5.20	3	37.50	7.37
12770837772...	0,,10268,00.html?target...	0.22	1	5.20	3	37.50	4.17
12770837772...	0,,10268,00.html	2.40	1	5.20	3	37.50	46.15

Figure 3 Figure showing creation of percent of page duration variable

The macro also extracts information about the type of browser used for accessing the page. This is a simple programming statement using SAS character functions. Similarly Date and Time variables are created from the Datetime stamp.

Id	Page_Name	Browser_Type	Browser_Version	Date	Time
127708377724...	0,,10268,00.html	Mozilla	5.0	02JAN2011	15:02:24
127708377724...	0,,10268,00.html	Mozilla	5.0	02JAN2011	15:02:49
127708377724...	?target=http%3A...	Mozilla	5.0	02JAN2011	15:03:38
127708377724...	0,,10268,00.html...	Mozilla	5.0	02JAN2011	15:04:10
127708377724...	0,,10268,00.html...	Mozilla	5.0	02JAN2011	15:04:36
127708377724...	0,,10268,00.html...	Mozilla	5.0	02JAN2011	15:04:59

Figure 4 Figure showing creation of browser and date time variables

The new dataset contains information that can be reported and used for various statistical analyses. This dataset can further be modified easily to a form where it can be used for building sequence models using SAS Enterprise Miner.

FILTERS – WEB ROBOTS

Multiple filters need to be applied to processed web log datasets prior to doing any kind of analysis on the data. One of the most important filters would be exclusion of web robots from the SAS dataset. Web Robots are machine-generated search engines that provide necessary service to sites like Google and the other search engines by providing fast access to the internet resources[7]. Access to resources is possible by creating a world wide index of available information. Identification and removal of robot becomes the vital part when activities like reporting the web site metrics is to be done. Variety of methods is used for removing robots; important ones of them is including user agent string exclusion. Usage of user agent string with the conjunction of IP addresses exclusion list could be one of the best ways to remove web robots. Sometimes, just using IP exclusion list may not solve the purpose as Internet Servers and IP addresses keep on changing. Code for some of the important exclusion list is mentioned in Appendix.

SUMMARY REPORT

The other important function of the macro is to generate relevant reports using the processed data. Reports help in answering various questions related to the website and visitor behavior like:

- Which is the most visited web page?
- Where are the visitors spending most of the page?
- Which is the most frequent exit page?
- What is the average time spent by a visitor on a particular page?

The processed data set can be used to answer these types of questions. The macro currently generates only basic reports. The macro can be modified to generate different types of reports according to the analyst's requirements.

The reports that the macro generates are:

- **Top Ten visited pages**
- **Top Ten web pages where visitors spent most of the time**
- **Number of pages visited on a daily basis**
- **Top ten exit pages**

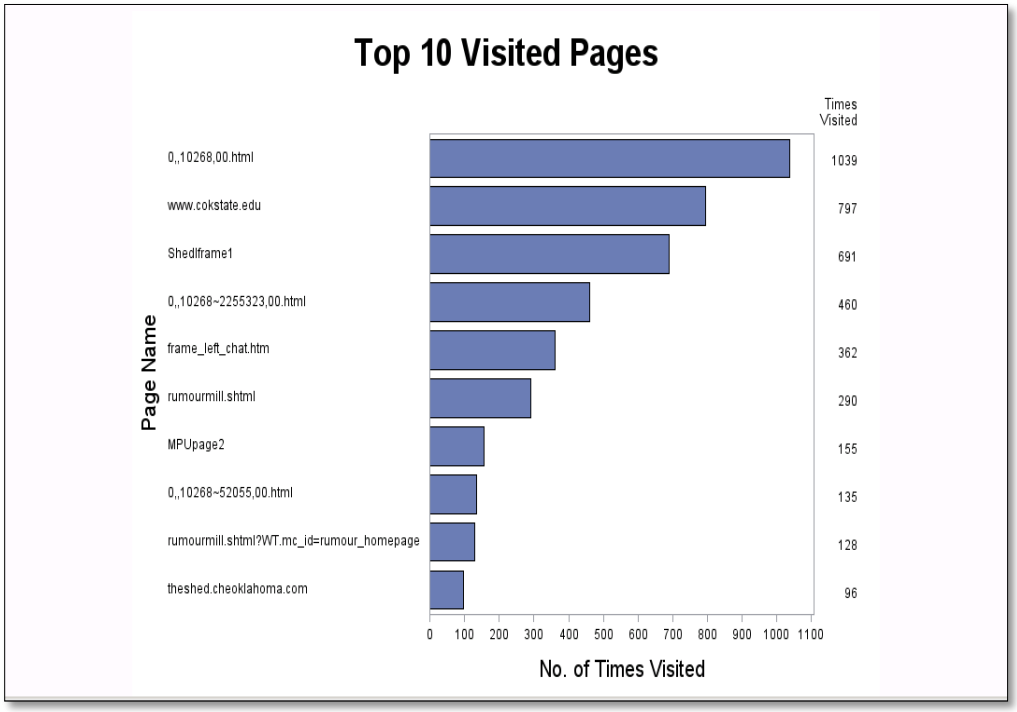

Figure 4 Top ten visited pages

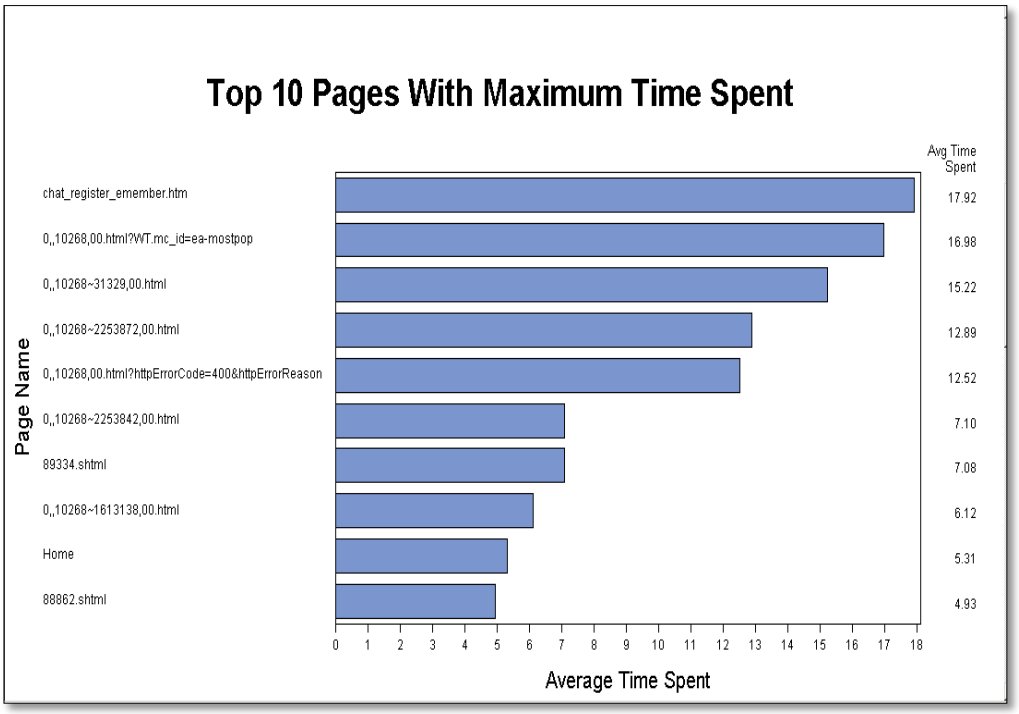

Figure 5 Top 10 Pages with maximum time spent

.

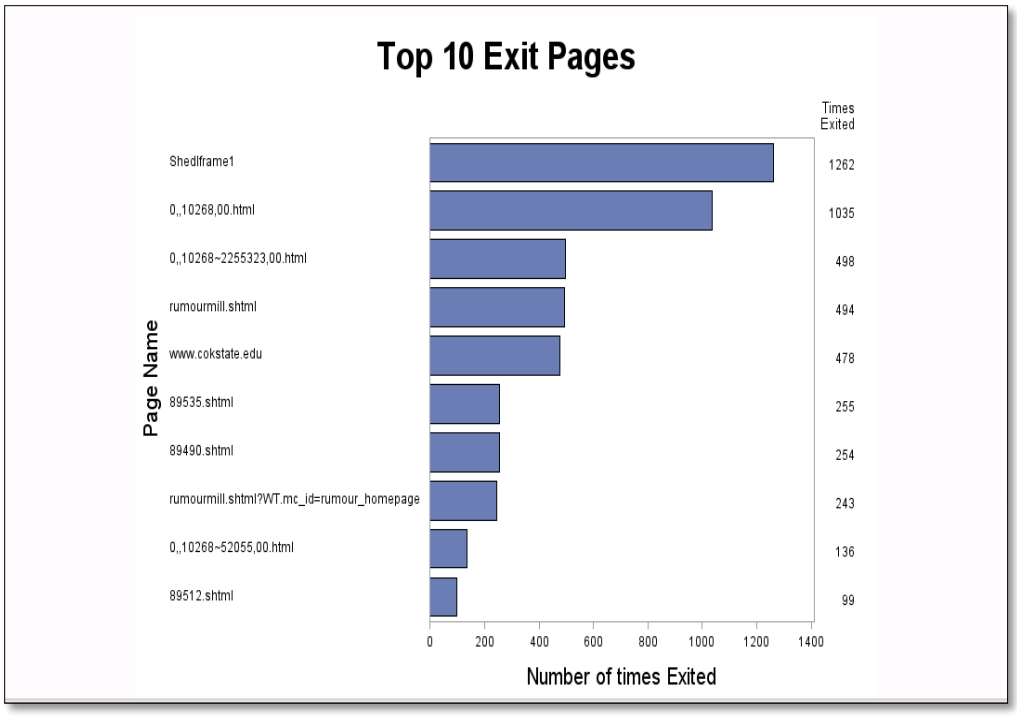

Figure 7 Top ten Exit pages

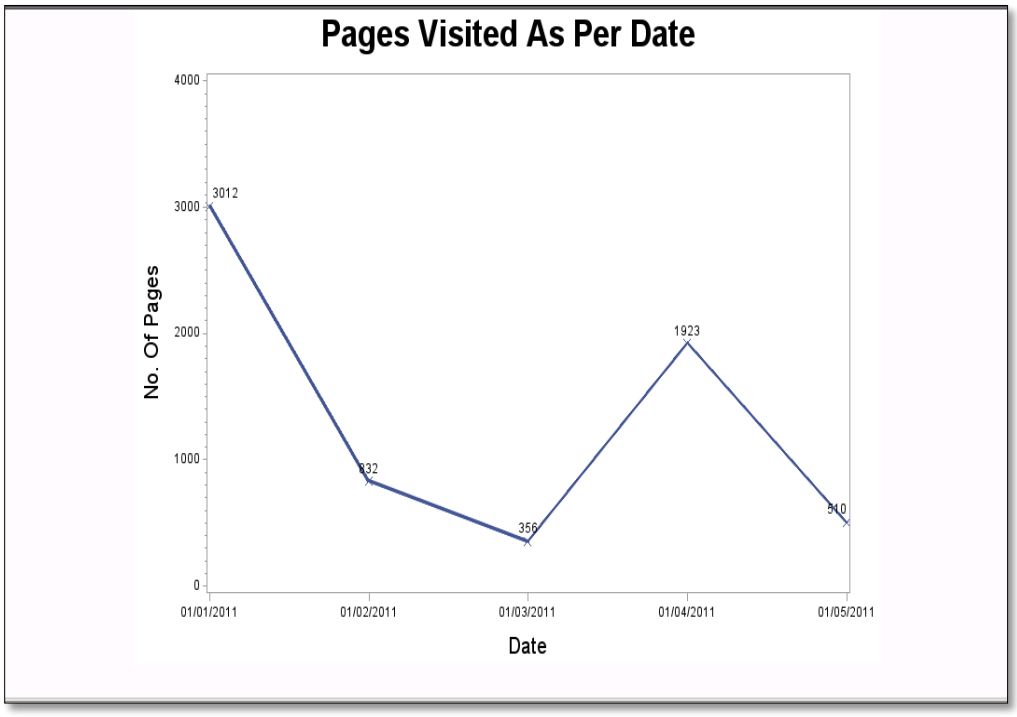

Figure 8 No. Of pages per day

CONCLUSION

Clickstream data has a lot of valuable information about web site visitor's online behavior. However the server log data are not available in the right format for analysis. SAS programming can be used to prepare data in a form that can be reported and used for various modeling analysis. The macro discussed in this paper can be easily used to prepare a SAS dataset from server access logs and generate basic reports. More sophisticated reports can be obtained by using any commercial web analytics applications that charge a lot of money (such as Adobe) or do not give researchers control over their data (such as Google Analytics).But, the SAS data set that is created by this free macro gives more control to analysts in terms of applying wide range of advanced analytics techniques and defining customized variables. This macro can also be customized by uses to include more reporting capabilities. We hope many users can use this free macro and tweak it to create SAS data sets from their own web logs and the apply sophisticated analytic techniques on those SAS data set.

REFERENCES

[1] Randolph E. Bucklina and Catarina Sismeirob "Advances in Clickstream Data Analysis in Marketing", JOURNAL OF INTERACTIVE MARKETING

[2] Avi Goldfarb, "Analyzing Website Choice Using Clickstream Data", Joseph L. Rotman School of Management University of Toronto

[3]Alan L. Montgomery, Shibo Li, Kannan Srinivasan, and John C. Liechty, "Modeling Online Browsing and Path Analysis Using Clickstream Data"

[4] Peter I. Hofgesang and Wojtek Kowalczyk, " Analysing Clickstream Data: From Anomaly Detection to Visitor Profiling" Free University of Amsterdam, Department of Computer Science,
Amsterdam, The Netherlands

[5]Wei Wang, "Parsing Web Logs with Base SAS®", Highmark Blue Cross Blue Shield, Pittsburgh, PA

[6] Kim Weller "Mainstreaming Web Data with SAS® Web Analytics 5.3", SAS Institute, Inc., Cary, NC

[7] Jenine Eason and Jerry Johannesen "CREATING MEANINGFUL DATA FROM WEB LOGS USING BASE SAS®", Autotrader.com, Peachtree City, GA

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Sumit Sukhwani
Oklahoma State University
E-mail: sumit.sukhwani@okstate.edu

Satish Garla
Oklahoma State University
E-mail: satish.garla@okstate.edu

Dr. Goutam Chakraborty
Oklahoma State University
E-mail: goutam.chakraborty@okstate.edu

APPENDIX

MACRO CODE

```
/*************************************************************************
Macro Name: create_webdata
Purpose:     Macro to create SAS  dataset from raw weblogs and generate PDF report
Inputs:      The macro uses two keyword parameters.

       Data = specify the name of the tab delimited input .txt file
       Infile = specify the path to the directory where data is available

*************************************************************************/

%macro create_webdata(data=,infile=);

Libname lib '&infile';

data &lib..&data. ;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */

infile '&in_file.\&data..txt' delimiter='09'x  MISSOVER DSD lrecl=32767 ;

    informat visid_high   best32.;
    informat visid_low    best32.;
    informat date_time    $50.;
    informat ip           $15.;
    informat browser      $246.;
    informat page_url     $246.;
    informat page_url1    $246.;

    format   visid_high         best32.;
    format   visid_low          best32.;
    format   date_time          $50.;
    format   ip                 $15.;
    format   browser            $246.;
    format   page_url           $246.;
    format   page_url1          $246.;

input visid_high visid_low  date_time ip$ browser page_url $ page_url1    $;

    run;

/* Creation of Id, DateTime and Browser related variables */

  data hit_data1;
  set &lib..&data.;

  informat Id best32.;
  format id best32.;
  format datetime datetime16.;
  length page_name  $50.;
  delim='/(;';
  Id=(strip(put(visid_high,best32.)) || strip(put(visid_low,best32.)));
  datetime=input(date_time,ANYDTDTM19.);
  page_name=coalescec(strip(scan(page_url,1,delim,'M')),strip(scan(page_url,-
  2,delim,'M')));
  browser_type=scan(browser,1,delim);
  browser_ver=scan(browser,2,delim);
  browser_med=scan(browser,3,delim);
```

```
/*Checking occurrences of robots*/

if index(lowcase(page_name), 'cyberworld/map') or
   index(lowcase(page_name), 'tmp') or
   index(lowcase(page_name), 'foo.html')or
   index(lowcase(page_name), 'keynote') or
   index(lowcase(page_name), 'libwww')or
   index(lowcase(page_name), 'msiecrawler')
then delete;


drop visid_high visid_low browser page_url1 delim date_time;
run;

/*Sort the data set in the reverse order(time) for each visitor.
This helps in easily calculating the time_spent per page*/

proc sort data=hit_data1;
by id descending datetime;
run;

data hit_data2;
set hit_data1;
by id;
retain end_time 100;

if first.id
then
       do;
       end_time=datetime;
       time_spent=0;
       end;
else
       time_spent=(end_time-datetime)/60;
end_time=datetime;
drop end_time;
run;

/*Sort the data set back to original form*/

proc sort data=hit_data2;
by id datetime;
run;

/*calculate the session based on time_spent. A new data set session_dur is also
created which captures session duration. This data set is later joined with the main
data set*/

data hit_data3 (drop=session_dur tmspent_flg session_start) session_dur(keep= id
session session_dur);
set hit_data2;
by id;
retain session 1 tmspent_flg 0 session_start 10000;

if first.id
then
       do;
              session=1;
              session_start=datetime;
       end;
if tmspent_flg ge 30
then
```

```
	do;
	session=session+1;
	session_start=datetime;
	end;
if last.id or time_spent ge 30
then
	do;
		session_dur=(datetime-session_start)/60;
		output session_dur;
	end;
tmspent_flg=time_spent;
output hit_data3;
run;
proc sort data=hit_data3;
by id session;
run;



/*Identify entrance page and exit page for each session for a visitor */


data ent_page(keep=id session ent_page) ext_page(keep=id session ext_page);
length ent_page $256.;
length ext_page $256.;
set hit_data3;
by id session;
delim='/(;';

if first.session
then
	do;
	ent_page=coalescec(strip(scan(page_url,-1,delim,'M')),strip(scan(page_url,-
2,delim,'M')));
	output ent_page;
	end;
if last.session
then
	do;
	ext_page=coalescec(strip(scan(page_url,-1,delim,'M')),strip(scan(page_url,-
2,delim,'M')));
	output ext_page;
	end;
run;


/*Create a table with number of times a page is visited by each visitor.
This data set is later joined with the main data set*/

proc sql;
create table times_visit as
select id, session, page_name,  count(*) as times_visited
from hit_data3
where page name ne ''
group by 1,2,3;
quit;

/*Join session_dur data set with the main data set */

proc sql;
```

```
create table hit_data4 as
select a.*, b.session_dur
from hit_data3 a left join session_dur b
on a.id=b.id and a.session=b.session;
quit;

/*Join times_visit data set with the main data set */

proc sql;
create table hit_data5 as
select a.*, b.times_visited
from hit_data4 a left join times_visit b
on a.id=b.id and a.session=b.session and a.page_name=b.page_name;
quit;

/*Join ent_page and ext_page data sets to create page_details data set */

proc sql;
create table page_details as
select a.*, b.ext_page
from ent_page a, ext_page b
where a.id=b.id and a.session=b.session;
quit;

/*Join page_details (entrance page, exit page) data set with the main data set */

proc sql;
create table hit_data6 as
select a.*, b.ent_page, b.ext_page
from hit_data5 a left join page_details b
on a.id=b.id and a.session=b.session;
quit;

/*Create data set to capture the number of pages visited per session. This count is
used in calculating percentages */

proc sql;
create table pages_per_session as
select id, session, count(*) as pages
from hit_data6
group  by 1,2;
quit;

/*Join the pages_per_session data set with the main data set*/

proc sql;
create table hit_data7 as
select a.*, b.pages
from hit_data6 a left join pages_per_session b
on a.id=b.id and a.session=b.session
order by id,datetime;
quit;

/*Create the final data set with all the remaining variables*/

data &lib..hit_data_final;
set hit_data7;

format pct_page_visit 8.2;
format pct_page_duration 8.2;
format time_spent 8.2;
format session_dur 8.2;
format Date date9.;
```

```
format time time.;

Date=datepart(datetime);
Time=timepart(datetime);
days_ago=date()-date;

if (pages ne 0)
then
pct_page_visit = (times_visited/pages)*100;
if (session_dur ne 0)
then
       pct_page_duration= (time_spent/session_dur)*100;
else
       pct_page_duration=0;
drop page_url pages datetime;
run;

/*Create Data sets for generating Report*/

proc sql outobs=10;
create table freq_pages as
select page_name, count(page_name) as Times
from research.hit_data_final
group by 1
order by 2 desc;
quit;

proc sql outobs=10;
create table time_spent as
select page_name,count(*) as count,sum(time_spent) as total_time,
calculated total_time/calculated count as avg_time_spent
from research.hit_data_final
group by page_name
order by avg_time_spent desc;
quit;

proc sql outobs=10;
create table exit_page as
select ext_page,count(*) as count
from research.hit_data_final
group by ext_page
order by count desc;
quit;

proc sql;
create table dateplot as
select date_new, count(*) as count
from research.hit_data_final
group by 1;
quit;

/*Define ODS Layout and generate PDF Report*/

ods listing close;
ods pdf file="H:\final\ClickStream.pdf" STARTPAGE=NO BOOKMARKGEN=NO;

axis1 label=(angle=90 height=15pt "Page Name")  minor=none;
axis2 label=(height=15pt "No. of Times Visited") minor=none;
proc gchart data=freq pages;
title2 height=25pt 'Top 10 Visited Pages';
hbar page_name/ sumvar= Times DESCENDING  sumlabel='Times Visited' raxis=axis2
maxis=axis1 coutline=black
woutline=1 outside=SUM;
```

```
run;

axis1 label=(angle=90 height=15pt "Page Name")  minor=none;
axis2 label=(height=15pt "Average Time Spent") minor=none;
ods region x=5.5 in y=0.3 in height=3.5 in width=5 in;
proc gchart data=time_spent;
title2 height=25pt 'Top 10 Pages With Maximum Time Spent';
hbar page_name/ sumvar= avg_time_spent DESCENDING sumlabel='Avg Time Spent'
raxis=axis2 maxis=axis1 coutline=black
woutline=1;
run;

axis1 label=(angle=90 height=15pt "Page Name")  minor=none;
axis2 label=(height=15pt "Number of times Exited") minor=none;
proc gchart data=exit_page;
title2 height=25pt 'Top 10 Exit Pages';
hbar ext_page/ sumvar= count DESCENDING  sumlabel='Times Exited' raxis=axis2
maxis=axis1 coutline=black
woutline=1;
run;

goptions reset=all;
SYMBOL1
       INTERPOL=JOIN
       POINTLABEL
       HEIGHT=10pt
       VALUE=X
       LINE=1
       WIDTH=2

       CV = _STYLE_;
Axis1
label=(angle=90 height=15pt "No. Of Pages");

Axis2
label=(height=15pt "Date");
PROC GPLOT DATA = dateplot;
title2 height=25pt 'Pages Visited As Per Date';
PLOT count * DATE_NEW  / VAXIS=AXIS1 HAXIS=AXIS2;
RUN; QUIT;
ods pdf close;
ods listing;

%mend;

%create_webdata(data=hit_data,infile=\\stwfile06.ad.okstate.edu\susers3\sukhwan\Research\hit_data.txt);
```