Module 2

This code performs two main tasks:

- Unzips and processes flat file server log data and creates a SAS dataset
- Creates a unique identifier

Unique identifier is created from three variables: Business userid, Cookie, Visitor Id, in order of priority. If Business userid is present the Unique identifier is Business userid, else it falls back to Cookie. If Cookie is not present, it further falls back to Visitor Id.

For example, given session of a visitor with 10 page visits, Business userid can be present in only one of the page visits. If Business userid is found in any 10 page visits, the code will populate Business userid across all 10 rows of session using Cookie/SessionId. Further using Cookie, Business userid is replicated across all sessions. Hence Business userid becomes the Unique identifier here.

If Business userid is not found, then Cookie is used as Unique identifier. If Cookie is also missing, Unique identifier is populated with Visitor Id.

If Visitor Id is also missing, Unique identifier is populated with Session Id.

```
%Macro Infor;
 informat date yymmdd10. ;
        informat time time20.3 ;
        informat c_ip $14. ;
        informat cs_username $1. ;
        informat cs_host $16. ;
        informat cs_method $3. ;
        informat cs_uri_stem $5000. ;
        informat cs_uri_query $5000. ;
        informat sc_status best32. ;
        informat sc_bytes $1. ;
        informat cs_version $1. ;
        informat cs_User_Agent_ $154. ;
        informat cs_Cookie_ $75. ;
        informat cs_Referer_ $300. ;
        informat dcs_geo $213. ;
        informat dcs_dns $36. ;
        informat origin_id $33. ;
        informat dcs_id $31. ;
        format date yymmdd10. ;
        format time time20.3 ;
        format c_ip $14. ;
        format cs_username $1. ;
        format cs_host $16. ;
        format cs_method $3. ;
        format cs_uri_stem $5000. ;
        format cs_uri_query $5000. ;
        format sc_status best12. ;
        format sc_bytes $1. ;
        format cs_version $1. ;
        format cs_User_Agent_ $154. ;
        format cs_Cookie_ $75. ;
        format cs_Referer_ $300. ;
        format dcs_geo $213. ;
        format dcs_dns $36. ;
        format origin_id $33. ;
        format dcs_id $31. ;
%Mend Infor;
%Macro read;
     input
                date
                time
                c_ip $
                cs_username $
                cs_host $
                cs_method $
                cs_uri_stem $
                cs_uri_query $
                sc_status
                sc_bytes $
                cs_version $
                cs_User_Agent_ $
                cs_Cookie_ $
                cs_Referer_ $
                dcs_geo $
                dcs_dns $
                origin_id $
                dcs_id $
     ;
%Mend read;

Filename ts pipe 'ls
/sasdata/[redacted]/*2015-*.log*';
Libname Web '/sasdata[redacted]';

  Proc sql noprint;
    Select Compress(path) into :path from sashelp.vslib where libname='WORK';
```

```
Quit;
        data input_files;
            infile ts End=final;
            input;
            file_name = _infile_;
            Call sYmput(Compress('Flname'||Put(_N_,3.)),Strip(File_name));
            P = Strip(scan(file_name,-1,'.'));

            Call symput('Type'||Compress(Put(_N_,3.)),'0');
            if p = 'gz' Then call symput(compress('type'||Put(_N_,3.)),'1');
            If Final Then Call symput('Total',_N_);
        run;

%Macro  FlatFileProcess;
            %DO I = 1 %TO &Total;
                %Let Flpath=&&Flname&i;
                %Put FlPath=&FlPath;
                %if &&Type&i = 1 %Then %Do;
                    x "gunzip -c &FlPath >  %Sysfunc(Compress(&path))/F&i..log";
                %End;
                %Else %Do;
                    X "cp &FlPath %Sysfunc(Compress(&path))/F&i..log";
                %End;
                Data Test;
                    Infile "%Sysfunc(Compress(&path))/F&i..log" delimiter = ' ' MISSOVER DSD
lrecl=32767  firstobs=2 obs=MAX;
                    Length Wt_co_f $50.WT_vt_sid $70. Wt_VtID $80. Wt_DcsVid $80. ;
                    %Infor;
                    %read;
                     if _N_ = 1 then  do;
                          retain ExpressionID ExpressionID1 ExpressionID2 ExpressionID3;

                          /* The i option specifies a case insensitive search. */
                          /*pattern = '/tagversion[^ ]*&cam=\d*&/i';*/
                          pattern = '/WT.co_f=[^ ]*?(&|WT\.)/i';
                          ExpressionID = prxparse(pattern);
                          Pattern1 = '/WT.vt_sid=[^ ]*?(&|WT\.)/i';
                          ExpressionID1 = Prxparse(Pattern1);
                          Pattern2 = '/WT.vtId=[^ ]*?(%26|&|WT\.)/i';
                          ExpressionID2 = Prxparse(Pattern2);
                          Pattern3 = '/WT.dcsvid=[^ ]*?(&|WT\.)/i';
                          ExpressionID3 = prxparse(pattern3);
                       end;

                       call prxsubstr(ExpressionID,Cs_Uri_Query, position, length);
                       if position ^= 0 then  do;
                          Wt_Co_f = substr(Cs_Uri_Query, position, length);
                          /*put match:$QUOTE. "found in " Cs_Uri_Query:$QUOTE.;*/

                          Drop Pattern ExpressionID;
                       End;
                       Position=0;
                       call prxsubstr(ExpressionID1,Cs_Uri_Query, position, length);
                       if position ^= 0 then  do;
                          WT_vt_sid = substr(Cs_Uri_Query, position, length);
                          /*put match:$QUOTE. "found in " Cs_Uri_Query:$QUOTE.;*/

                          Drop Pattern1 ExpressionID1;
                       End;
                       Position=0;
                       call prxsubstr(ExpressionID2,Cs_Uri_Query, position, length);
                       if position ^= 0 then  do;
                          WT_vtid = substr(Cs_Uri_Query, position, length);
                          Drop Pattern2 ExpressionID2;
                       End;
                       Position=0;
                       call prxsubstr(ExpressionID3,Cs_Uri_Query, position, length);
                       if position ^= 0 then  do;
```

```
                    Wt_dcsvid = substr(Cs_Uri_Query, position, length);
                    Drop Pattern3 ExpressionID3;
                  End;
            Drop Pattern: Expression: Position: Length:;
             id = _N_;
            wt_co_f = compress(Scan(Strip(Scan(Wt_co_f,2,'=')),1,'&%'));
            Wt_vt_sid = compress(Scan(strip(scan(wt_vt_sid,2,'=')),1,'&%'));;
            wt_vtid = Compress(Scan(Strip(scan(wt_vtid,2,'=')),1,'&%'));;
            wt_dcsvid=Compress(Scan(strip(scan(wt_dcsvid,2,'=')),1,'&%'));;
            if Missing(Wt_Co_F) And Not Missing(Wt_Vt_Sid) Then Do;
                WtPop='1';
                SsnStrtTime = Reverse(Scan(Strip(reverse(Wt_vt_sid)),1,'.'));
                Wt_Co_F= Compress(Tranwrd(Wt_vt_sid,"."||Strip(SsnStrtTime),""));
            end;
            Drop SsnStrtTime;
            Run;
            X rm "%Sysfunc(Compress(&path))/F&i..log";
            Proc Append Base=Web.WebLogData Data=Test Force;
            Run;
            Proc Datasets Lib=work Nolist;
                Delete Test;
            Run;
            Quit;
        %End;
%Mend FlatFileProcess;
options obs=Max;
%FlatFileProcess;

Proc Sort Data=Web.WebLogData;
    by Wt_Co_F ;
Run;


Proc format;
value $miss
' ' = 'Missing'
other = 'NM';
quit;

Proc Freq data=Web.WebLogData;
Table Wt_dcsvid*Wt_Vtid*Wt_co_f*WtPop*Wt_vt_sid / list missing;
format Wt_dcsvid Wt_Vtid Wt_co_f Wt_vt_sid $miss.;
Run;


Proc Sort Data=Web.WebLogData(Keep=Wt_Co_F Wt_Vt_Sid Wt_DcsVid Id) Out=DcsID;
by Wt_DcsVid;
Where Not missing(Wt_DcsVid) and Wt_Dcsvid ne 'null';
Run;


Proc Sort Data=DcsID(Keep=Wt_Co_F Wt_DcsVid) Out=UniCof Nodupkey;
by Wt_Co_F;
Where not missing(Wt_Co_F);
Run;


Data NMDcsVid MissDcsVid(Drop = DcsVid);
    Merge Web.WeblogData(In=table1) UniCof(In= Table2 Rename=(Wt_Dcsvid=DcsVid));
    By Wt_co_F;
    If Table1 & Table2  Then Do;
         Fltype = 'D';
         Output NMDcsVid;
    End;
    Else Do;
        If Not (missing(Dcsvid) and missing(wt_dcsvid) and missing(wt_vtid) and missing(wt_co_f) ) Then
Do;
            Output MissDcsVid;
```

```
        End;
    End;
Run;


Data NmVtid;
    set MissDcsVid(Keep=Wt_Vtid Wt_co_F);
    If Not missing(Wt_VtId);
Run;


Proc Sort Data=NmVtid(Keep=Wt_Vtid Wt_Co_f) Out=nmvtid1 Nodupkey;
    by Wt_vtid Wt_co_f;
    Where Not missing(Wt_Co_f);
Run;

Proc Sort Data = NmVtid1 Nodupkey;
 By Wt_co_F;
Run;

Data PopDcsVid(Drop=DcsVid1) MissDcsVid(Drop=DcsVid);
    Merge MissDcsVid(In=table1) NmVtId1(In = Table2 Rename = (Wt_vtid=DcsVid));
    By Wt_Co_F;
    If Table1;
    if Table1 & Table2 Then Do;
        FlType ='V';
        Output PopDcsVid;
    End;
    Else Do;
        DcsVid1 = Wt_Co_F;
        Output MissDcsVid;
    End;
Run;


Proc Sort Data=MissDcsVid(Keep=Wt_Co_F DcsVid1 ) Out=Co_final Nodupkey;
    by Wt_Co_F;
    Where Not missing(Wt_Co_F);
Run;

Data MissDcsVid;
    Merge MissDcsVid(In=Table1) Co_final(In=Table2 Rename= (DcsVid1=DcsVid));
    By Wt_Co_F;
    If Table1;
    If table1 & Table2 Then Do;
        FlType='C';
    End;
Run;

Data WeblogData;
Attrib DcsVid Wt_co_f WT_vt_sid date time c_ip cs_uri_stem cs_uri_query
 cs_Referer_ cs_Cookie_ cs_User_Agent_ cs_host cs_method cs_username
 cs_version Fltype WtPop Wt_DcsVid Wt_VtID dcs_dns dcs_geo dcs_id id
 origin_id sc_bytes sc_status Label  = '';

    Set NMDcsVid PopDcsVid MissDcsVid;
        If _N_ = 1 Then do;
            retain ExpressionID ;
            pattern = '/WT.vtvs=[^ ]*?(&|WT\.)/i';
            ExpressionID = prxparse(pattern);
        End;
        If Missing(Wt_Vt_sid) Then Do;
         call prxsubstr(ExpressionID,Cs_Uri_Query, position, length);
            if position ^= 0 then  do;
                Wt_Vt_Sid = Compress(Scan(substr(Cs_Uri_Query, position, length),2,'='));
                Drop Pattern ExpressionID;
                Sid='1';
            End;
```

```
        Drop Position Length;
       End;
Run;

Proc Sort Data= WeblogData Out=Web.WeblogData;
    By DcsVid Wt_co_f Date Time Wt_Vt_Sid;
Run;

Proc Freq data=WeblogData;
    Table FlType*DcsVid*Wt_dcsvid*Wt_Vtid*Wt_co_f*Wt_vt_sid / list missing;
    format Wt_dcsvid Wt_Vtid Wt_co_f DcsVid Wt_vt_sid $miss.;
Run;
```