Paper 140-2011

Using SAS® to Help Fight Crime: Scraping and Reporting Inmate Data

William G. Roehl, Capella University, Minneapolis, MN

ABSTRACT

A colleague posed an interesting question recently regarding the makeup of inmates incarcerated within the Dakota County, Minnesota jail system. While limited summary data is made available by the county every year in clunky Microsoft Word® tables, the information provided would not help to answer the question. Fortunately the county also provides a web interface to view inmates currently incarcerated in the Dakota County Jail. With informative data including ethnicity, gender, age, arrest date and city being made available to the public, what better way to collect and report on data to answer this question but by pulling directly from the source and not having to rely solely on infrequently updated summary tables? The only problem is that the very rich data on the Dakota County Inmate Search website is not in a format conducive to reporting and thus some massaging must occur to make it useful for this purpose.

This paper illustrates how SAS® can be used to create extremely detailed and informative reports from data available on the Internet displayed in a manner not typically meant for reporting (i.e. not in flat files or other easily importable data files). Using only SAS® functions and no external software, this paper will demonstrate how SAS® is able to download, parse, and report on data from over 7,100 inmate records of those incarcerated over the last year and a half in Dakota County, Minnesota and answer not only the question which prompted this investigative work but also many others as well.

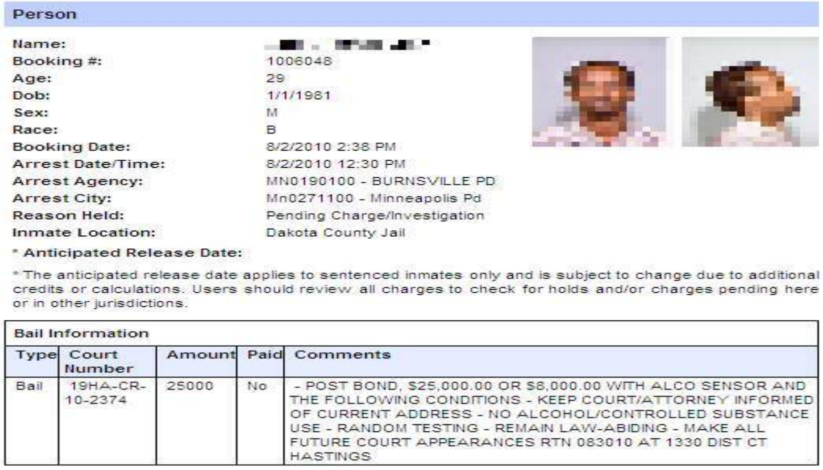
The intended audience for this paper is SAS® developers with a working knowledge of SAS®, basic understanding of regular expressions, and a desire to integrate publicly available data into their own research.

Code was developed with SAS® 9.1.3 running under Microsoft Windows® XP Professional.

INTRODUCTION

Raw data is available everywhere on the Internet and sometimes it comes in extremely handy flat files such as comma or tab separated variable formats. Unfortunately in many cases this raw data is usually displayed in a format which is not conducive to reporting and would normally be very difficult to manually copy and paste, reformat, and then report. However, SAS® has a variety of tools which make it easy to automatically download these web pages, parse the data contained within, and store the resultant information in datasets which then allow for reporting and saving as any format imaginable, including flat files.

DAKOTA COUNTY INMATE DATA ON THE WEB



Person

Name:	[redacted]
Booking #:	1006048
Age:	29
Dob:	1/1/1981
Sex:	M
Race:	B
Booking Date:	8/2/2010 2:38 PM
Arrest Date/Time:	8/2/2010 12:30 PM
Arrest Agency:	MN0190100 - BURNSVILLE PD
Arrest City:	Mn0271100 - Minneapolis Pd
Reason Held:	Pending Charge/Investigation
Inmate Location:	Dakota County Jail
* Anticipated Release Date:	

* The anticipated release date applies to sentenced inmates only and is subject to change due to additional credits or calculations. Users should review all charges to check for holds and/or charges pending here or in other jurisdictions.

Bail Information

Type	Court Number	Amount	Paid	Comments
Bail	19HA-CR-10-2374	25000	No	- POST BOND, $25,000.00 OR $8,000.00 WITH ALCO SENSOR AND THE FOLLOWING CONDITIONS - KEEP COURT/ATTORNEY INFORMED OF CURRENT ADDRESS - NO ALCOHOL/CONTROLLED SUBSTANCE USE - RANDOM TESTING - REMAIN LAW-ABIDING - MAKE ALL FUTURE COURT APPEARANCES RTN 083010 AT 1330 DIST CT HASTINGS

Display 1. Sample Dakota County inmate website information screen

This is a subset sample of the data made available for each and every inmate currently incarcerated in the Dakota County, Minnesota jail. The county has provided plenty of interesting data points which, when taken in aggregate, can be used in reporting datasets to build informative reports and charts which could drive interesting insights into the crime witnessed in the county. By having more insight into the incarcerated population within the county one might be able to answer questions about public safety funding requirements, provide an informative profile of the general inmate population, or even allow for better cross-departmental collaboration by showing the dynamic nature of where crime happens and at what time of the year.

With all these enticing applications the problem still exists where data, such as in the example above, would need to be copied and then pasted for each individual inmate and then placed in a file for later parsing. Using the FILENAME function with the 'url' option, it is possible to save this webpage as a text document within SAS® for processing with regular expressions.

Dakota County provides a list of those who are currently incarcerated within the jail. Using that list it is possible to find a beginning booking number (seen below as "1006048") and thereafter these booking numbers can be incremented by one to provide the next available inmate in the system.

Example:

```
filename dakota url "http://co.dakota.mn.us/InmateSearch/Details.aspx?PIN=1006048";
```

Subset of downloaded HTML file:

```
<table  id="tblPerson" style="width:100%" cellspacing="0" cellpadding="0">
        <tr>
          <td class="textBold">Booking #:</td>
          <td class="text"><span id="ctl00_ContentPlaceHolder1_lblBookingNumber">1006048</span></td>
        </tr>
```

There are two macros which do the majority of the work which includes reading from the inmate website and parsing the data:

The first is the %PROCESSDATA macro. It handles the repetitive task of parsing chunks of the data which are read from the input file into only the data we want (e.g. name, booking number, age, gender, etc).

%PROCESSDATA

Parameter	Description
LINE	The data read from the input file
VARNAME	Field name to be attached to the data parsed from the file

Example:

```
        patternID = prxparse('/lblBookingNumber/');
        if prxmatch(patternID, _infile_) then do;

                %processData(_infile_,booking_num); match = 1;

        end;
```

The %PROCESSDATA macro uses some simple regular expressions to parse the pertinent data out of the data read from the input file (_INFILE_) and adds the parsed data back out to a SAS® data set in the 'new' field with the chosen name of the field stored in 'id'. Eventually, when the dataset is transposed with the TRANSPOSE procedure the data will appear under a field named by 'id'.

Example dataset created with %PROCESSDATA:

patternID	New	id	match
11	1006048	booking_num	1

Table 1. %PROCESSDATA macro table output

After this the rest of the %GETINMATEDATA macro continues to process and eventually transposes the data, adds some more reporting fields, and recodes another before outputting the individual inmate to a larger reporting dataset of all the inmates downloaded via the APPEND procedure.

The second is the %GETINMATEDATA macro. This macro finds the relevant data within the input file and passes it along to the %PROCESSDATA macro for parsing. It then transposes the resultant data to put each inmate's data on one line, recodes race into a human readable ethnicity label, and creates an age range field for more useful summary reporting before appending the individual inmate into a single larger reporting dataset of all the inmates downloaded and parsed.

%GETINMATEDATA

Parameter	Description
BOOKING_NUM	The inmate's booking number (format: 10000000)

Example:

```
data _null_;
       %getinmatedata(1006048);
       %getinmatedata(1006049);
       %getinmatedata(1006050);
       %getinmatedata(1006051);
       %getinmatedata(1006052);
run;
```

Example reporting dataset created (subset):

booking_num	Name	age	age_range	dob	sex	race	eth	booking_dt
1006048	[withheld]	29	25 to 29	1/1/1981	M	B	Black	8/2/2010 2:36 PM

Table 2. %GETINMATEDATA macro table output

Using this data is now as simple as summarizing with a few DATA steps, SQL procedure code blocks, and creating some charts with the GCHART procedure.

NOTES

It is also important to note that the code used in this example has also been modified to pull data from two other crime data sources on the web: Scott County's Inmate Registry and Minnesota's Level 3 Sex Offender database. Both of these sources required very little modification to work with this code even though they are utilizing completely different data display formats. With this in mind it only makes sense that someone with an understanding of regular expressions would be able to easily extract the necessary information from other websites for their own reporting purposes.

REFERENCES

- Dakota County Inmate Search (http://services.co.dakota.mn.us/InmateSearch/)

CONTACT INFORMATION

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

William Roehl
Capella University
225 S 6th St, 9th Fl
Minneapolis, MN 55402
william.roehl@capella.edu
http://www.capella.edu

DAKOTA COUNTY INMATE DATA SOURCE CODE

```
/*********************************************************************************\
PROGRAM INFORMATION
Project : SGF2011 Submission: Dakota County Inmate Data
Purpose : Download, parse and analyze inmate registry data
Inputs  : http://services.co.dakota.mn.us/InmateSearch/Details.aspx?PIN=xxxxxxx
Outputs : dataset

PROGRAM HISTORY
2010-10-11 WR Initial program developed.
\********************************************************************************/;

/*********************************************************************************\
Macro to parse HTML and provide the name of the field based on passed parameters
\********************************************************************************/;
%macro processData(line,varName);
    /* Data is in the same format, strip out what is not needed for each item */
    new = prxchange('s/^.*<td class="text"><span
                     id="ctl00_ContentPlaceHolder1_lbl.*">//',-1,&line);
    new = prxchange('s/<\/span><\/td>.*$//',-1,new);
    new = prxchange('s/,/ /',-1,new);

    /* Name the field */
    id = "&VarName";
%mend processData;

/*********************************************************************************\
Macro pulls down the individual inmate's HTML file, searches for portions of the
file, and sends it off to be parsed
\********************************************************************************/;
%macro getInmateData(BOOKING_NUM);
    /* Pull individual inmate data */
    filename dakota url
    "http://services.co.dakota.mn.us/InmateSearch/Details.aspx?PIN=&BOOKING_NUM"
    lrecl=2000;

    /* Scrape the inmate data using regular expressions */
    data scraped(where=(match=1));

           /* Relevant data doesn't start until about 80 lines down in the file */
           infile dakota firstobs=80;
           input;

           patternID = prxparse('/lblBookingNumber/');
           if prxmatch(patternID, _infile_) then do;
                 %processData(_infile_,booking_num); match = 1;
           end;

           patternID = prxparse('/lblName/');
           if prxmatch(patternID, _infile_) then do;
                 %processData(_infile_,name); match = 1;
           end;

           patternID = prxparse('/lblAge/');
           if prxmatch(patternID, _infile_) then do;
                 %processData(_infile_,age); match = 1;
           end;

           patternID = prxparse('/lblDob/');
           if prxmatch(patternID, _infile_) then do;
                 %processData(_infile_,dob); match = 1;
           end;

           patternID = prxparse('/lblSex/');
           if prxmatch(patternID, _infile_) then do;
                 %processData(_infile_,sex); match = 1;
```

```
       end;

       patternID = prxparse('/lblRace/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,race); match = 1;
       end;

       patternID = prxparse('/lblBookingDate/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,booking_dt); match = 1;
       end;

       patternID = prxparse('/lblArrestDate/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,arrest_date); match = 1;
       end;

       patternID = prxparse('/lblArrestAgency/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,arrest_by); match = 1;
       end;

       patternID = prxparse('/lblArrestCity/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,arrest_city); match = 1;
       end;

       patternID = prxparse('/lblReasonHeld/');
       if prxmatch(patternID, _infile_) then do;
              %processData(_infile_,reason_held); match = 1;
       end;
run;

/* We want the data on one line */
proc transpose data=scraped(keep=new id) out=scraped_oneline(drop=_NAME_);
       var new;
       id id;
run;

/* Create dataset with fields in proper order, add a few for reporting */
data inmate;
       /* force the order of the dataset */
       format booking_num $7.
                 name $100.
                 age $3.
                 age_range $11.
                 dob $10.
                 sex $1.
                 race $1.
                 eth $10.
                 booking_dt $20.
                 arrest_date $20.
                 arrest_by $50.
                 arrest_city $50.
                 reason_held $50.;

       set scraped_oneline;

       /* Create an age range based on the scraped "age" */
       if (age < 20)              then age_range = "0 to 19";
       if (age >= 20 && age < 25) then age_range = "20 to 24";
       if (age >= 25 && age < 30) then age_range = "25 to 29";
       if (age >= 30 && age < 35) then age_range = "30 to 34";
       if (age >= 35 && age < 40) then age_range = "35 to 39";
       if (age >= 40 && age < 45) then age_range = "40 to 44";
       if (age >= 45 && age < 50) then age_range = "45 to 49";
       if (age >= 50 && age < 55) then age_range = "50 to 54";
```

```
           if (age >= 55)             then age_range = "55 and Over";

           /* Recode the scraped race into a human readable ethnicity */
           if (race = "A") then eth = "Asian";
           if (race = "B") then eth = "Black";
           if (race = "I") then eth = "Indigenous";
           if (race = "U") then eth = "Unknown";
           if (race = "W") then eth = "White";
           if (race = "")  then eth = "Unknown";
    run;

    /* Save the inmates into one big dataset for reporting */
    proc append base=dakota_inmates data=inmate force; run;
%mend getInmateData;

/*********************************************************************************\
Pull the data down for a selection of inmates
\*********************************************************************************/;

/* Fetch some inmates and store them in a single dataset for reporting */
data _null_;
    %getinmatedata(1006048);
    %getinmatedata(1006049);
    %getinmatedata(1006050);
    %getinmatedata(1006051);
    %getinmatedata(1006052);
run;

/*********************************************************************************\
Summarize the data for reporting
\*********************************************************************************/;

/* Age ranges */
proc sql;
  create table AgeRangeSummary as
  select distinct age_range, count(age) as count
  from dakota_inmates
  group by age_range
  ;
quit;

/* Arrest city */
proc sql;
create table ArrestCitySummary as
select scan(arrest_city,-1,"-") as agency, count(name) as inmates
from dakota_inmates
where (upcase(arrest_city) like '%BURNSVILLE PD'
    or upcase(arrest_city) like '%FARMINGTON PD'
    or upcase(arrest_city) like '%HASTINGS PD'
    or upcase(arrest_city) like '%ROSEMOUNT PD'
    or upcase(arrest_city) like '%EAGAN PD'
    or upcase(arrest_city) like '%APPLE VALLEY PD'
    or upcase(arrest_city) like '%LAKEVILLE PD')
group by arrest_city
order by scan(arrest_city,-1,"-")
;
quit;

/* Arrest day */
proc sql;
  create table ArrestsByDaySummary as
  select distinct
       compress(put(input(scan(arrest_date,-3," "),mmddyy10.),EURDFDWN.)) as Day
     , count(name) as arrested
  from dakota_inmates
  group by calculated day
  ;
quit;

/**********************************************************************************\
Plot the data
\*********************************************************************************/;

TITLE 'Age Ranges';
PROC GCHART DATA=AgeRangeSummary;
      HBAR count;
RUN;
QUIT;

TITLE 'Arrests by Day';
PROC GCHART DATA=ArrestsByDaySummary;
      HBAR arrested;
RUN;
QUIT;

TITLE 'Gender';
PROC GCHART DATA=dakota_inmates;
      PIE sex / DISCRETE VALUE=INSIDE
                PERCENT=NONE SLICE=OUTSIDE;
RUN;
QUIT;

TITLE 'Gender';
PROC GCHART DATA=dakota_inmates;
      PIE eth / DISCRETE VALUE=INSIDE
                PERCENT=NONE SLICE=OUTSIDE;
RUN;
QUIT;

TITLE;
```

Scraping and Reporting Jail Registry Data