

**Paper PO23**

## **SSN validation – Virtually at no cost**

Milorad Stojanovic  
RTI International  
Education Surveys Division  
RTP, North Carolina

### **ABSTRACT**

Using SSNs without validation is not the way to ensure quality of the data you collect, merge in, or use from secondary sources. Verification of SSNs is out of the scope of this paper.

The author's goal was to find an easy but still powerful way to validate a given SSN. He used a publicly available source (site of the Social Security Administration) to get current 'high group' numbers. He developed a program / macro which can be easily used by each user. Also, he points out scenarios in which validating SSNs is not advantageous.

### **INTRODUCTION**

The usage of Social Security Numbers (SSNs) in society is very broad. For example, when seeking to obtain some state and federal documents, enrolling in a post secondary education system, applying for loans, opening bank accounts, and perhaps more importantly, when applying for a job, we are asked to present our SSN. That said those examples are just the tip of iceberg called SSN. From the very beginning the usage of SSNs has been abused to some degree (either intentionally or unintentionally). In recent years such abuse has even lead to 'Identity Theft' in some cases, though the legal aspect of that is quite out of the scope of this paper. The author doesn't have such intention and moreover doesn't have any formal law education.

Mainly the author is working with surveys which are in the area of education and Medicare and, in some of the surveys, the respondents were asked to present their SSN.

In some of the aforementioned studies, SSNs in conjunction with other identifying information were used as a key for matching survey data with other previously collected data (in our arrangement or by other independent organizations).

In theory, a SSN is a very useful piece of data for merging together data from various sources. It is unique for citizens and permanent residents of the US, it is relatively compact, and it is managed by one government organization; the Social Security Administration (SSA).

### **ABOUT SSN AND VALIDATION**

#### **VALIDATION:**

Validation means the Social Security Number (SSN) is a valid number. Validation can be, and usually is, determined by a mathematical calculation that determines that the number MAY BE A VALID NUMBER, along with the state and year in which that the number MAY HAVE BEEN ISSUED. This process in no way ensures that the number: has actually been issued, does not belong to a deceased person, or even that number actually was issued by the Government to your candidate. In this type of process, there is no actual access to real data from the Social Security Administration (attempted or implied). In an employment situation, an SSN Validation will not provide any useful information about the actual status of an applicant's Social Security Number.

#### **VERIFICATION:**

Verification means that the Social Security Administration (SSA) actually confirms that the number has been issued to a particular person, the state of issue has been provided by the SSA, and the year of issue has been provided by the SSA. While the SSA will not confirm to anyone whom the SSN is actually issued to, the top of your report will be imprinted with a starred box declaring "possible fraud" if the name on file with the SSA is not the exact name presented in your request. The report is also imprinted if, for example, the SSN has already had death benefits paid on it or if the person is drawing SSI disability payments. There are a total of 17 classes of fraud.

SSNs consist of nine digits organized in three groups usually divided by a hyphen ("-").

Let say SSN = 123-45-6789 as an example. The first three digits define AREA, the middle two digits define GROUP, and the last four digits define SERIAL NUMBER.

**Area** code ranges from 001 to 732 as of the date this paper was written. An area code of 000 is invalid.

The following sequence is used by the SSA to assign **Group** numbers:

Odd numbers, 01 to 09  
Even numbers, 10 to 98  
Even numbers, 02 to 08  
Odd numbers, 11 to 99  
Group codes of "00" aren't assigned

The **serial** number can't be 0000.

When you get an SSN from a respondent, how should you go about verifying whether the SSN is valid or invalid? There are, in the author's opinion, three possible levels of verification (including validation).

1. Full checking which requests access to SSA databases (with active and inactive SSNs).  
In that case you are able to check more data on a person in addition to the SSN.
2. Validation which indicates whether an SSN is formally correct.  
You will be able to determine if the respondent gave an invalid or still non-issued SSN.
3. A third quick and 'dirty' check will give results less accurate than those obtained using option 2 but will require less effort. It is a simple program which will look for obvious inconsistencies in content of the SSN.

There is of course a fourth way of verifying and/or validating an SSN, however it does cost some money. In the US there are several tens of commercial organizations or companies that would do it for you but with a price tag of \$0.25 (one quarter) per case or higher. Some companies charge more and some less but there is no one company that will do it for free (at least the author is not aware of such a company).

If you need to check data for let's say 100,000 cases, it could be significant financial burden (the total cost would be at least \$25,000 US).

On the other side if you have survey data (data collected based on interviewing respondents and not on official documents) you have the potential of losing some cases if you are attempting to do a complete check of the SSN.

Moreover you would still be in position to collect perfectly valid SSNs that may belong to one of the respondent's parents (if, for example, the respondent is a High School student) or to someone in his or her family.

Should we collect SSNs at all? The answer should be NO if we do not plan to link data collected in the survey with some other sources of data that are already available for the same person (e.g. FAFSA data etc.). If we need to match our data with some other sources of data then we must collect information from respondents that will allow us to do it later and, as a rule, it is the SSN.

From the author's experience he would like to point out that some 'financial' institutions that lend money to students accept, to certain degree, SSNs that are obviously invalid or even bogus. One potential source of problems is using Tax Identification Numbers (TIN) as a surrogate for SSN. Many foreign students get TINs because they are not eligible for a 'real' SSN (the author's own children serve as an example).

The future of using SSNs is not quite clear to the author. He thinks that respondents will become more and more reluctant to provide SSNs as more and more cases of "Identity Theft" surface (1). On the other side, a recent report on identity theft gave some optimistic signs by identifying less cases where "Identity theft" resulted from using electronic devices (computers, Internet, etc.) than from classic methods (using discarded but not shredded bank statements, cancelled checks etc.). Also some SSNs were reserved for display in wallets or purses (2). The most misused SSN of all time is (078-05-1120). In 1938, wallet manufacturer the E. H. Ferree Company in Lockport, New York decided to promote its product by showing how a Social Security card would fit into its wallets. A sample card, used for display purposes, was inserted in each wallet. In the peak year of 1943, 5,755 people were using that number. In all, over 40,000 people reported this as their SSN.

In 1940 the Board published a pamphlet explaining a new program which showed a facsimile of a social security card on the cover. The card in the illustration used a made-up number of 219-09-9999.

Let's go back to the primary topic of this paper which is how to perform some level of validation of SSNs while not spending a lot of money and still ending up with acceptable results?

1. Comparing SSNs and other respondent's data with SSA database information.
2. Utilizing a more complex macro that uses live data from the SSA's website (public data).
3. Utilizing a simple macro for formal validation of SSNs.

The first option is out of the scope of this paper. If you are able to spend a quarter (\$0.25) per case or so, you will find a number of commercial firms who will complete the task for you. Again, you should have a guarantee that your data will not be lost or disseminated without your knowledge by the company you choose to validate your SSNs.

To perform validation at a certain level of quality the author developed a program (macro) in SAS® which draws in a reputable source of public information – website of the Social Security Administration.

### PROGRAM / MACRO FOR SSN VALIDATION

```
%macro Validation_SSN (Input_DataSet=, SSN=, SSN_Flag=, __Final_Check=) ;
%local Input_DataSet SSN SSN_Flag __Final_Check ;
* Connection to SSA site with current High group data. This information is
updated once per month. ;
FILENAME ssa_hgrp URL 'http://www.ssa.gov/employer/highgroup.txt' ;
proc format ;
value flg 0 = 'Invalid' 1 = 'Valid';
run ;
DATA __high_group(keep=area group seqno) ;
INFILE ssa_hgrp lrecl=256 pad ;
INPUT @1 text1 $char256. ;
if substr(text1,1,1) in (' ', 'A', 'T', 't', 'N', '09'x) then delete ;
text1 = compress(translate(text1, " ", "092A"x));
length area group seqno 3 ;
do i = 1 to 6 ;
    area = input(substr(text2, (i-1)*5 + 1, 3), 3.) ;
    group = input(substr(text2, (i-1)*5 + 4, 2), 2.) ;
    odd = mod(group, 2) ;
    if odd = 1 and group <= 9 then seqno = (group + 1) / 2 ;
    else if odd = 0 and (group >= 10 and group <= 98) then seqno = group / 2 + 1
;
    else if odd = 0 and (group >= 2 and group <= 8) then seqno = group / 2 + 50
;
    else if odd = 1 and (group >= 11 and group <= 99) then seqno = (group - 1) /
2 + 50 ;
    if area > 0 then output ;
end;
run ;
data __Area_Group_Seqno ;
do i=0 to 999; area = i ; group = 0 ; seqno = 0 ; output ; end ;
drop i ;
run ;
data __High_Group_All ;
merge __Area_Group_Seqno(in=in1) __high_group ;
by area ;
if in1 ;
run ;

data &__Final_Check.(COMPRESS=YES) ;
/* Repository of the max issued group number converted to seqno */
array Agrp{0:999} 3 _temporary_ ;
do until(allareas) ;
```

```

        set __High_Group_All end = allareas ;
        Agrp(area) = seqno ;
    end ;
    do until(allssns) ;
        set &Input_DataSet. end = allssns ;
        area1 = input(substr(&ssn., 1, 3), 3.) ;
        group1 = input(substr(&ssn., 4, 2), 2.) ;
        odd1 = mod(group1, 2) ;
        if group1 = 0 then &SSN_Flag. = 0 ;
        else do ;
            if odd1 = 1 and group1 <= 9 then seqno1 = (group1 + 1) / 2 ;
            else if odd1 = 0 and (group1 >= 10 and group1 <= 98) then seqno1 = group1 /
/ 2 + 1 ;
            else if odd1 = 0 and (group1 >= 2 and group1 <= 8) then seqno1 = group1 /
2 + 50 ;
            else if odd1 = 1 and (group1 >= 11 and group1 <= 99) then seqno1=(group1
- 1)/2+50 ;
            if Agrp{area1} >= seqno1 then &SSN_Flag. = 1; /* Valid SSN */
            else &SSN_Flag. = 0; /* Invalid SSN */
        end ;
        length __first_digit $ 1 ;
        __first_digit = &ssn. ;
        output &__Final_Check. ;
    end ;
    stop ;
    drop area group seqno ;
run ;

options nocenter nonumber nodate ;
ods rtf file="C:\SESUG07\report.rtf" style=default ;
proc freq data=&__Final_Check. ;
title "&__Final_Check. - freqs on valid and invalid SSNs" ;
title2 "&sysdate." ;
tables __first_digit * &SSN_Flag. / missing list nocum ;
format &SSN_Flag. flg. ;
run ;
ods rtf close ;

proc datasets library = work ;
* Removing unnecessary data sets. ;
delete __High_Group_All __Area_Group_Seqno __high_group ;
run ;
%mend Validation_SSN ;

```

For testing purposes, the RANUNI function was used and 250,000 'SSN' numbers were generated. Those numbers were submitted to Validation\_SSN macro and the results are presented in Tab 1.

```

*Test run of macro ; *Input data set ;
data ssn_chk ;
length SSN $ 9 ;
do i = 1 to 250000 ;
    rn_num = ceil(999999999*ranuni(i)) ;
    ssn = put(rn_num, Z9.) ;
    output ;
end ; drop i ;
run ;

%Validation_SSN(Input_DataSet=ssn_chk, SSN=ssn, SSN_Flag=ssn_flag,
_Final_Check=Out_File)

```

Result from testing is in Tab 1.

Tab 1.

| <b>__first_digit</b> | <b>ssn_flag</b> | <b>Frequency</b> | <b>Percent</b> |
|----------------------|-----------------|------------------|----------------|
| <b>0</b>             | <b>Invalid</b>  | 13119            | 5.25           |
| <b>0</b>             | <b>Valid</b>    | 12133            | 4.85           |
| <b>1</b>             | <b>Invalid</b>  | 13072            | 5.23           |
| <b>1</b>             | <b>Valid</b>    | 11806            | 4.72           |
| <b>2</b>             | <b>Invalid</b>  | 6170             | 2.47           |
| <b>2</b>             | <b>Valid</b>    | 18687            | 7.47           |
| <b>3</b>             | <b>Invalid</b>  | 10269            | 4.11           |
| <b>3</b>             | <b>Valid</b>    | 14838            | 5.94           |
| <b>4</b>             | <b>Invalid</b>  | 4359             | 1.74           |
| <b>4</b>             | <b>Valid</b>    | 20466            | 8.19           |
| <b>5</b>             | <b>Invalid</b>  | 3662             | 1.46           |
| <b>5</b>             | <b>Valid</b>    | 21415            | 8.57           |
| <b>6</b>             | <b>Invalid</b>  | 15269            | 6.11           |
| <b>6</b>             | <b>Valid</b>    | 9851             | 3.94           |
| <b>7</b>             | <b>Invalid</b>  | 23564            | 9.43           |
| <b>7</b>             | <b>Valid</b>    | 1559             | 0.62           |
| <b>8</b>             | <b>Invalid</b>  | 25006            | 10.00          |
| <b>9</b>             | <b>Invalid</b>  | 24755            | 9.90           |

When you are reading results from the Tab1 please keep in mind few things. First, those results are from sample based on random number generator which simulates real SSNs. Second there is no guarantee sample SSNs covers uniformly whole spectrum of area codes starting with 0, 1, etc. Word 'Invalid' means real SSN was not issued yet. When you are looking at the very last two rows it is obvious that all SSNs are invalid. No one SSN was issued with first digit 8 or 9 yet. Higher percentage of 'Invalid' SSNs in the row with 7 as a first digit means that smaller portion of SSNs for all area codes starting with 7 was issued so far.

Author likes to add few words on applied algorithm and program. Applied algorithm is relatively simple but on the other side it is very efficient. On the stand alone PC with Windows XP and SAS V 9.1.3 author checked up to 60,000 cases per second. Result could vary from PC to PC.

## CONCLUSION

Validation of SSNs is possible with virtually no cost. You don't need to pay unnecessarily for a service that provides additional data about an individual that you will not use at all. The SAS® program or macro is straightforward and could be done in a relatively small amount of time. The base source of data (high group numbers) is official and reliable.

## FURTHER READING

Author recommends for further reading paper (7). It is an excellent paper from Carnegie Mellon University.

## ACKNOWLEDGEMENTS

I would like to thank Christopher Alexander and Robert Lee from the Education Studies Division (RTI) for all their help, comments and support in producing this paper.

## REFERENCES:

1. Speech by SEC Staff: Compliance Professionals versus Identity Thieves by <http://www.sec.gov/news/speech/2006/spch100506jhw.htm>
2. Social Security Cards Issued by Woolworth, <http://www.ssa.gov/history/ssn/misused.html>
3. Your Social Security Number And Card, SSA Publication No. 05-10002, October 2006 (Recycle prior editions), ICN 451384, [http://www.pueblo.gsa.gov/cic\\_text/fed\\_prog/ssnumber/ssnumber.htm](http://www.pueblo.gsa.gov/cic_text/fed_prog/ssnumber/ssnumber.htm)
4. SSN Validation Web Service <http://www.codeproject.com/aspnet/ssnvalidator.asp>
5. Social Security Number Identification Verification Search (40 free searches) [http://www.usinfosearch.com/Free\\_ssn\\_search.htm](http://www.usinfosearch.com/Free_ssn_search.htm)
6. United States Social Security Number Validation (using other programming languages than SAS®) <http://www.quentinsagerconsulting.com/documents/10013.htm>
7. SSN Validation as a Way to Combat Identity Theft <http://privacy.cs.cmu.edu/dataprivacy/projects/ssnwatch/identitytheft.html>
8. Validation (SSN), JavaScript, <http://webscripts.softpedia.com/script/Forms-and-Controls-C-C/Validation-SSN--21924.html>
9. SSN VALIDATION PROCEDURES, State of Pennsylvania [http://www.dpw.state.pa.us/oimpolicymanuals/manuals/bop/supp/950/950\\_C.htm](http://www.dpw.state.pa.us/oimpolicymanuals/manuals/bop/supp/950/950_C.htm)

## DISCLAIMER

All opinions and suggestions stated in the paper on how to validate SSNs do not necessarily reflect the opinions and suggestions of the Education Survey Studies Division (RTI). Using any of commercial websites or services mentioned in references for checking SSNs is at your fully responsibility.

## CONTACT INFORMATION

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

Milorad Stojanovic  
Education Surveys Division  
RTI International  
3040 Cornwallis Rd  
RTP, NC, 27909  
Work Phone : (919) 541-7376  
E-mail: [milorad@rti.org](mailto:milorad@rti.org)

## TRADEMARK INFORMATION

SAS® and all other SAS® Institute Inc. product or service names are registered trademarks or trademarks of SAS® Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.