

# Using Google to Solve SAS® Problems

Michael Todd, Nth Analytics, Flemington, NJ

## ABSTRACT

SAS® has a long history as the language for statistical programming in the pharmaceutical industry. In the past 30+ years, most, if not all, of the problems statistical programmers face on a day-to-day basis have been solved by somebody. Usually there is a paper somewhere on the Internet about it. Using Google queries, statistical programmers can leverage the vast SAS knowledge base on the Internet, and get quick answers to a variety of problems.

## INTRODUCTION

SAS has been around longer than 30 years. Most of the SAS programming problems have been worked out somewhere, by somebody. To solve a problem, it is often a matter of finding the solution online, rather than working through the problem by yourself. You can greatly enhance your programming skills by leveraging the knowledge of others.

We all get used to doing things a certain way. Many problems can be solved using code already available at our companies. However, if new situations come up, there may not be anything to copy from. If new technologies get implemented, we probably will need some help just to get them working. Training does not cover all the situations. We need an ongoing source of knowledge. With Google, there is always someone to ask for help.

In this paper, I approach the programming issues from the viewpoint of a consultant who works for many organizations in the pharmaceutical industry, including Contract Research Organizations (CROs). These companies have many different ways of doing things. Usually, they are not anxious to change. As a 'guest in the house', you need to do things their way, and quickly. This can require quick mastery of unfamiliar techniques.

I explore using Google to solve five very different programming problems, and explain my reasons for choosing the particular solution I did.

- Simulating the lag function in PROC SQL
- Methods of putting 'Page x of y' on RTF output
- Age in months calculation
- Confidence intervals for the median
- International encoding difficulties opening a dataset

All of the solutions returned by the searches are valid. It is question of which solution is most easily implemented, cleanest, and best suited to a particular organization.

## GOOGLING SAS PROBLEMS: OVERVIEW

There are five main sources of information available online for solving SAS problems:

- **SAS documentation:** a great source of answers to SAS problems. Documentation for Base SAS, all procedures, ODS, and most other topics are available at <http://support.sas.com>
- **SAS books** are available through Google's controversial projects to make all books available online in a searchable format. This is handy when you are looking for a solution to a single problem, and don't need the whole book.
- **Social media:** SAS Institute has a strong commitment to social media. Up-to-date resources, including Twitter, Facebook, and YouTube feeds are available at <http://support.sas.com/community/socialmedia/index.html>.
- **SAS-L.** One of the original social media sites, founded in 1985, and still active. It is a listserv, which is an email server. If someone posts an email, it goes to everyone on the list, and people can respond in real time. Many expert programmers have answered questions over the years. Emails from the archives are available

from 1996 to the present, and often contain targeted solutions to SAS problems. The web address is <http://listserv.uga.edu/archives/sas-l.html>.

- **Conference papers** are a major source of SAS programming solutions. These papers are well-reasoned, peer-reviewed, and tend to be focused on a single topic. They often have step by step, simple solutions where one can lift the code straight out and use it.
- <http://www.lexjansen.com> often comes up on Google searches. This site has a collection of over 10,000 SAS conference papers.

## GOOGLE STORES YOUR SEARCHES

Each time you search, Google stores what you searched and the result. It also stores the number of times you searched it. You will need an account, and to be logged in on your Google account when searching. This tends to happen automatically. If you forget how solved a problem, you can go to <https://www.google.com/history>, log in, and search your searches. As shown in Table 1, I often search PROC SQL. I have difficulty grasping some of the syntax. In the age of Google, this is not a major problem, because I can easily find the answer

Table 1

Most Frequent PROC SQL Searches	
PROC SQL noprint	PROC SQL except operator
SAS PROC SQL correlated subquery	PROC SQL into separated by
PROC SQL into	SAS PROC SQL intersect
SAS SQL coalesce	SAS PROC SQL begins with operator
PROC SQL insert	PROC SQL like operator
SAS SQL between operator	SAS PROC SQL retain
SAS PROC SQL union syntax	SAS PROC SQL row difference

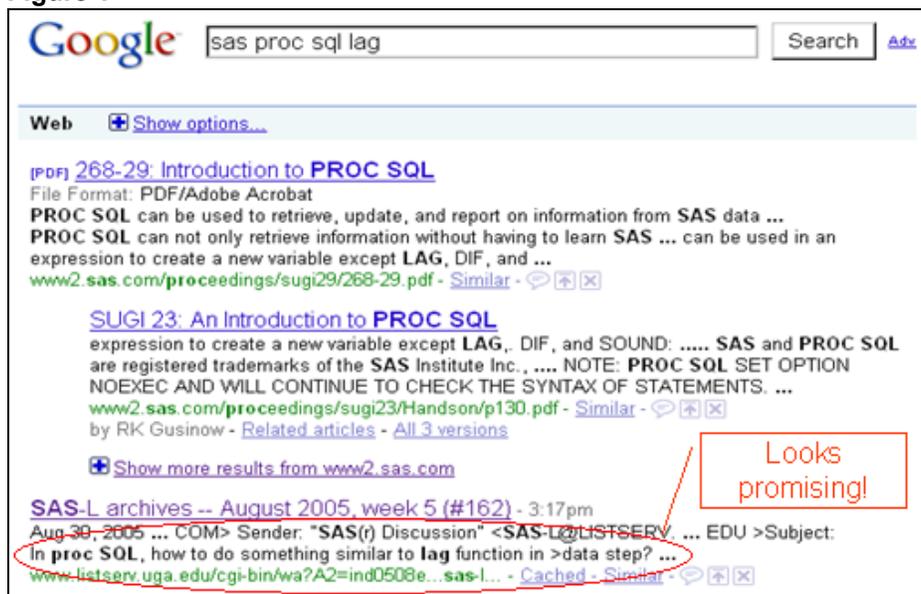
## SQL LAG FUNCTION PROBLEM

In this section I show how to use PROC SQL to simulate the lag function. Why would I want to do this? I could easily use the DATA STEP. This type of problem comes up more than one would expect in on-site consulting. With the advent of SAS Enterprise Guide, some companies are phasing out the DATA STEP. It is possible to consult in a place where the DATA STEP is discouraged.

## GOOGLE SEARCH: SAS PROC SQL LAG

In this case, the specific task was to use PROC SQL to simulate the lag function to find the cutoff for the 25<sup>th</sup> percentile. Figure 1 shows the results of the first search. In 2005, someone asked this question on SAS-L: "How do you do something similar to the lag function in a data step?"

Figure 1



Opening the link, as seen in Figure 2, it looks like a potentially good solution. The code is clean, and looks reasonably simple and straightforward to implement. There is a subquery that I don't totally understand, but looks like it will probably work.

**Figure 2 (Schreier, 2005)**

```
Here's an example. First use the LAG function in a DATA step.

data age_lag;
set sashelp.class;
age_lag = lag(age);
keep name age age_lag;
run;

To emulate that in SQL, first add a column with serial numbers for the rows:

data numbered;
set sashelp.class;
sn + 1;
run;

Then code a subquery to get the AGE value from the "previous" row:

proc sql;
create table age_lag_sql as
select class.name,
       class.age,
       (select lag.age
        from numbered as lag
        where class.sn - lag.sn = 1) as age_lag
from numbered as class
order by class.sn
;
```

It worked. As shown in Figure 3, I changed the variable names to match those in my programs, and was able to get the answer I wanted, the value of the 25th percentile. This is an example of a successful search.

**Figure 3**

```
data work.OneWayFreqs;
set work.OneWayFreqs2;
retain sn 0;
sn+1;
run;

proc sql;
create table work.OneWayFreqs_lag as
select class.param,
       class.window,
       class.base,
       class.cumFrequency,
       class.cumPercent,
       (select lag.CumPercent
        from work.OneWayFreqs as lag
        where class.sn - lag.sn = 1) as CumPct_lag,
       (select lag2.Base
        from work.OneWayFreqs as lag2
        where class.sn - lag2.sn = 1) as base_lag
from work.OneWayFreqs as class
order by class.cumPercent
;
quit;
```

BASE	cumFrequency	cumPercent	CumPct_lag	base_lag
46	1	0.40322581		
48	2	0.80645161	0.40322581	46
49	6	2.41935484	0.80645161	48
51	12	4.83870968	2.41935484	49
52	20	8.06451613	4.83870968	51
53	25	10.0806452	8.06451613	52
54	30	12.0967742	10.0806452	53
55	35	14.1129032	12.0967742	54
56	37	14.9193548	14.1129032	55
57	41	16.5322581	14.9193548	56
58	46	18.5483871	16.5322581	57
59	53	21.3709677	18.5483871	58
60	58	23.3870968	21.3709677	59
61	61	24.5967742	23.3870968	60
62	68	27.4193548	24.5967742	61
63	71	28.6290323	27.4193548	62
64	77	31.0483871	28.6290323	63

25<sup>th</sup> percentile cutoff

**RTF SEARCHES**

RTF is another thing I have difficulty remembering. With Google, this is not an issue. I can always search and find the answer. Over time, I have learned to do more targeted searches, particularly when I have searched the problem before. Table 2 shows the RTF topics I have searched:

Table 2

Most Frequent RTF Searches	
SAS RTF startpage	RTF font code
SAS ODS PROC REPORT RTF image	SAS RTF page x of y
SAS RTF hidden text	SAS ODS RTF page 1 of
SAS RTF margins	RTF escape character indent
SAS ODS RTF text=	RTF \nofpages
SAS RTF tags	SAS ODS RTF ^page
SAS RTF different style header and body	SAS ODS RTF title J=L
RTF underline	ODS RTF J=L J=R title on same line
SAS ODS RTF TOC	SAS ODS RTF escapechar ^left

The purpose of these searches is always to get the output to look exactly the way I want it. The need to do this tends to come up in the CRO industry. Clients may insist that output looks a certain way. However, I have also seen FDA reviewers make similar sorts of requests.

### GOOGLE SEARCH: SAS RTF PAGE X OF Y

One thing I have searched successfully is how to put 'Page x of y' (for example, Page 1 of 20, Page 2 of 20, etc.) on each page of the table or listing. I searched on **SAS RTF PAGE X OF Y**. The results are in Figure 4.

Figure 4

The screenshot shows a Google search interface with the following results and annotations:

- Result 1:** [Page X of Y with PROC REPORT](#) - 12:14pm. File Format: PDF/Adobe Acrobat - View. SAS Institute Inc. (b), "Q: In ODS RTF, can I get my page numbers in page X of Y format?." SAS FAQ # 4010. Product or Solution: Base SAS. ... changchung.com/download/pageXofY\_draft.pdf - Similar - [Icons]. by CY Chung - 2005 - Related articles. **Annotation:** Bad: macro-based
- Result 2:** [15727 - Writing PAGE X OF Y in RTF does not work with BODYTITLE](#) - 12:14pm. Raw RTF code or the special function PAGEOF can be used on a TITLE or FOOTNOTE statement to write PAGE X of Y text in RTF output. However, if the BODYTITLE ... support.sas.com/kb/15727.html - Cached - Similar - [Icons]. **Annotation:** Interesting, but not practical
- Result 3:** [alternate \(.sas\) - SAS Customer Support](#) - 12:14pm. The length of the special text must be long enough to hold the generated Page X of Y text. Does not apply to RTF files because the internal pagination is ... support.sas.com/md/base/ods/.../pageofpp\_public.sas - Cached - Similar - [Icons]. **Annotation:** Bad: too complicated
- Result 4:** [ODS RTF: Practical Tips](#) - 4 visits - 12:15pm. File Format: PDF/Adobe Acrobat - View. Since ODS RTF has been available in production in SAS .... "Page x of y" can be inserted into the footer using the following code: ... www.nesug.org/Proceedings/nesug03/at/at007.pdf - Similar - [Icons]. **Annotation:** Ok, but old
- Result 5:** [MeasureIT - Issue 5.11 - SAS 9 Tips - Part II](#). In SAS 9 if you want to add page numbers to your RTF output in the form of "page x of y" then you can use the in-line formatting by specifying an escape ... www.cmg.org/measureit/issues/mit46/m\_46\_1.html - Cached - Similar - [Icons]. **Annotation:** Winner!

There has been a lot of work done on this issue. I got five very relevant results. In Google searches, usually the first result is the best, because it has the most links to it. In this case, it was a matter of deciding which of the five solutions was best.

## SOLUTION 1: MACRO-BASED

The first result was not suitable for my purposes. It is macro-based. Usually macro-based solutions are not good, because they present too many validation issues. I always look for a PROC, a function, or some really simple code.

For this problem, all I want to do is put Page x of y in my title. In Figure 5, the macro is not shown. Figure 5 just shows how to call the macro. This is complicated enough! To call the macro, you to put a CALL EXECUTE inside a COMPUTE block and then pass the whole PROC REPORT in. This is clearly too complicated for what should be a fairly easy solution.

Figure 5 (Chung and Dunn, 2005)

```
/* example usage */
options linesize=64 nonumber nodate;

%pageXofY(report=%nrstr(

proc report data=one nowd headline missing;
column var;
define var / display;

compute after _page_;
call execute('%let page = %eval(&page. + 1);');
length _XofY $&len.;
_XofY = symget('page') || ' of ' || symget('pages');
line 'page ' _XofY $&len.;
endcomp;
run;

))
```

## SOLUTION 2: INTERESTING, BUT NOT PRACTICAL

The solution shown in Figure 6 uses PROC TEMPLATE in a rather elegant manner. PROC TEMPLATE can apply styles to any report. Used as a global template, this method could put Page x of y on every report in the same place. The problem was no one at this particular organization used PROC TEMPLATE. Although it was the best solution, it was too much change for the organization, just to do Page x of y.

An alternative is to use PROC TEMPLATE to write Page X of Y, which will place the pageof information in the upper right corner of each page. The code below will apply to the default page number location.

```
ods escapechar='^';

proc template;
define style styles.test;
parent=styles.rtf;
style pageno from pageno /
font=fonts("strongfont")
posttext=" of ^{lastpage} ";
end;
run;
```

Figure 6 (SAS Institute, 2005)

## SOLUTION 3: OVERLY COMPLEX FOR THE PROBLEM

The approach shown in Figure 7 is clearly way too complicated for this problem. It requires a lot of complex parameter checking before even starting to implement 'Page x of y'. Again, I am looking for ideally one line of code to solve this problem.

```
%pageofpp(infile="test1.pdf",outfile="test2.pdf",text=&pagnumt,type=pdf)

* Example 2 - Centered in a TXT file ;
%let pagnumt=[Page_00000_of_99999];
footnote1 "%sysfunc(repeat(%str( ),50))&pagnumt";
...
%pageofpp(infile="t1.txt",outfile="t2.txt",text=&pagnumt,type=txt,justify=c)

-----*/
%local parmerr
infilen outfilen
numpages numpagel
slength rtfcommand lpageof
;
%*-----
Validate parameters
-----;
%let parmerr=0;

%if (~%length(&infile)) %then %do;
%put ERROR: INFILE not specified.;
%let parmerr=1;
%end;
```

Figure 7 (SAS Institute, undated)

#### SOLUTION 4: TOO OLD

In Figure 8, the solution is perfectly valid. It is clean, easy to read, and requires only one line of code to implement. It is a way of embedding RTF field codes inside of SAS code. However, it mentions Word 97. That was a long time ago! I was hoping to find something a little newer.

Figure 8 (Tong, 2003)

"Page x of y" can be inserted into the footer using the following code:

```
footnote1 j=1 '{Page \field {\*\fldinst
PAGE \*\MERGEFORMAT}} { of \field
{\*\fldinst NUMPAGES \*\MERGEFORMAT}}';
```

Similarly, "Page y" can be inserted into the footer using the following code:

```
footnote1 j=1 '{ Page \field {\*\fldinst
NUMPAGES \*\MERGEFORMAT}}';
```

When the document is opened the page numbers may initially appear wrong, especially in Word 97. This is because Word has not refreshed the fields. Doing "Print preview" can solve this problem.

#### SOLUTION 5: WINNER

Figure 9 shows the best solution, given the particular organization that will use it. Although technically this requires an extra line of code to define the **ODS ESCAPECHAR**, this is as easy as it gets. Placing the ODS ESCAPECHAR in front of the automatic PAGEOF variable (available causes Page x of y to print on every page of the report. It is update-to-date, simple to read, simple to use, and very reliable.

Figure 9 (Mason, 2007)

##### SAMPLE CODE FOR RTF

```
ods escapechar = '\';
title 'This document will have page x of y '
      j=r 'Page \{pageof}' ;
ods rtf file='c:\test.rtf' ;
proc print data=sashelp.prdsale;
run;
ods rtf close;
```

#### AGE CALCULATION SEARCH

Most pharmaceutical companies and CROs have a standard macro to compute age. This is the standard formula:

```
ageyear=floor((vdate - birthdt + 1) /365.25 );
```

Age in months is required for pediatric studies (infants < 2 years of age). We can easily extend the standard formula to return age in months by multiplying by 12.

```
agemos=floor((vdate - birthdt +1)*12 /365.25);
```

Examining this code, it is easy to see that it is an approximation. The formula for years adjusts for leap years by dividing by 365.25 instead of 365. The formula for months further approximates that by multiplying by 12, assuming all months have an equal number of days. For validation, it would be interesting to try a different method.

#### GOOGLE SEARCH: SAS AGE IN MONTHS

This is another case where a lot of work has been done. As shown in Figure 10, there are several solutions to choose from.

Figure 10

The screenshot shows a Google search for "sas age in months". The search bar contains the text "sas age in months" and a "Search" button. Below the search bar, there are several search results. The first result is titled "Accurately calculating age in only one line" and includes a SAS macro definition: `%macro age(date,birth); floor ((intck('month',&birth,&date) - (day(&date) < day(&birth))) / 12) %mend age;`. The second result is titled "24808 - Calculating Age with Only One Line of Code" and also includes the same macro definition. The third result is titled "24567 - Calculate a person's age" and discusses determining a person's current age. The fourth result is titled "SAS FAQ - Date Processing with SAS Software" and includes a SAS code snippet: `DATA getage; birthday=25Dec1972'd; datevar=TODAY(); /* Divide months by 12 to get years */ age=INT(INTCK('MONTH',birthday,datevar)/12); /* Adjust age if ...`. The search results are displayed in a list format with "Results 1 - 10 of" shown at the top right.

In this case, the most authoritative answer would be from the SAS knowledge base on <http://support.sas.com>: **CALCULATING AGE WITH ONLY ONE LINE OF CODE**. That's what I want. This article presents the standard solution under 'what doesn't work'. What they mean is this is an approximation, and does not take into account calendar months. For this, you either need some complicated programming, or the INTCHK function. INTCHK gives you the interval between any two dates, in days, weeks, months, etc. Here we use it for months

Figure 11 (SAS Institute, 2004)

The screenshot shows SAS documentation for the INTCK function. It starts with the function signature: `intck('month',birth,somedate)`. The text explains that the function returns the number of times the first day of a month is passed between birth and somedate. An enhancement is needed to alter this into the number of times the same day of the starting month is passed. This is achieved by subtracting one month if the day number of somedate is earlier than the day number of birth. The code snippet shows: `intck('month',birth,somedate) - (day(somedate) < day(birth))`. The text states: "This now gives exactly the correct number of months for any pair of dates." A section titled "A one-line solution" shows the final macro definition: `age = floor ((intck('month',birth,somedate) - (day(somedate) < day(birth))) / 12);`

Substituting the variable names for my particular dataset, I used the following methods on a dataset with 3077 records from an integrated database of pediatric studies.

**Production Method:** `floor((VDATE - BIRTHDT +1)*12 /365.25)`

**Validation Method:** `intck('month',BIRTHDT, VDATE) - (day(VDATE) < day(BIRTHDT))`

Comparing the results, 68 of 3077 age calculations (2%) did not match. If we look at the results in detail in Figure 12, subject 10194, born on December 29, would reach his 3rd month birthday on March 29. The standard method computes the age as 2 months old. For such young infants, this is a big discrepancy.

Figure 12

Listing of Age (Months) Calculation Mismatches					
Subject Number	Date of Birth	Visit Date	Age (Months) - New Method	Age (Months) - Old Method	
101048	28APR2004	27NOV2006	✓ 30	31	Does not reach 31 months until 28NOV2006
101053	23JUN2005	22NOV2006	✓ 16	17	
101054	09JAN2006	08FEB2007	✓ 12	13	Reached 3 months on 29MAR2006
101090	10DEC2004	10MAR2006	✓ 15	14	
101094	29DEC2005	29MAR2006	✓ 3	2	
101138	13APR2006	11JAN2007	✓ 8	9	
101140	14APR2005	12JAN2006	✓ 8	9	
101140	14APR2005	13JAN2006	✓ 8	9	
101143	01APR2006	31JUL2007	✓ 15	16	
101197	02MAY2005	31JAN2007	✓ 20	21	
101232	06FEB2006	06SEP2006	✓ 7	6	
101233	19FEB2005	19SEP2006	✓ 19	18	
101236	03NOV2005	03APR2007	✓ 17	16	
101238	15AUG2005	14DEC2006	✓ 15	16	
300001	28DEC2004	27JAN2006	✓ 12	13	
300001	28DEC2004	28MAR2006	✓ 15	14	
300020	19AUG2003	18SEP2006	✓ 36	37	
300063	07AUG2004	06FEB2006	✓ 17	18	
300120	31OCT2005	31JUL2007	✓ 21	20	

## CONFIDENCE INTERVAL FOR A MEDIAN

Another unusual request I received as a consultant was to compute the confidence interval for the median, as part of the descriptive statistics summary. I have been a statistician or programmer in the pharmaceutical industry since 1981, and I was asked to provide this exactly once. Fortunately, I had Google to rely on. Otherwise, I had no idea how to do this.

## GOOGLE SEARCH 1: SAS CONFIDENCE INTERVAL MEDIAN

Once again, a surprising amount had been published on this topic. Although I was completely unfamiliar with the problem, several people worked on it. The search results are shown in Figure 13:

Figure 13

Google search results for "sas confidence interval median". The search bar shows the query and the search button. The results are displayed under the "Web" tab, showing 10 results out of approximately 179,000. The first two results are PDF documents from Lexjansen.com, both titled "Calculating a Nonparametric Estimate and Confidence Interval Using SAS". The third result is an article from pubstat.org titled "The Mean, Median, and Confidence Intervals of the Kaplan-Meier ...". The fourth result is an archived email from SAS-L archives dated January 31, 2001, discussing how to get SAS to return a median and 95% confidence intervals.

## FIRST RESULT

The title of the first result looked interesting: CONFIDENCE INTERVALS IN THE ANALYSIS AND REPORTING OF CLINICAL TRIALS. Additionally, it was NESUG paper, and therefore likely to contain SAS code I could use. Based on a quick read, however, the article did not look promising. Figure 14 summarizes my initial reaction:

Figure 14 (Guangbin, 2003)

<p>Not promising ...</p> <p>such cases, distribution-free confidence intervals should be constructed. A distribution-free interval sometimes may not exist, and its length is generally longer than the corresponding distribution-dependent interval for a particular distribution. This is the price that one pays for not making the distribution assumption (Hahn and Meeker, 1991). So, a distribution-dependent confidence interval should be chosen whenever there is solid evidence that the data follows a tractable distribution.</p>	<p>code. The following code using SAS version 6.09 represents one way to obtain the 95% confidence interval for the mean treatment difference from an ANOVA model.</p>	<p>Paper is from 2003, but they are using Version 6.09?</p>
<p><b>DISTRIBUTION-DEPENDENT CONFIDENCE INTERVAL</b></p> <p>If the assumption that the data are normally distributed is valid, one can construct confidence intervals for the mean treatment difference. The general form of a confidence interval for the mean difference between two treatment groups (Group A and Group B) is</p>	<pre> *-----* * The following statements get output * * datasets containing statistics * * needed for calculation of confidence * * interval * *-----* proc glm data=final outstat=glmdt   noprint;   class str1 s1str s1nved;   model s1dep=s1ndp;   lsmeans str1/pdiff stderr tdiff     out=lsmeandt;  *-----* * The following statements get the * * degree of freedom from the model * *-----* data _null_; set glmdt;   if _type_='ERROR';   call symput('df', df);  *-----* * The following statements get the * * LSMEANS from the model * *-----* data _null_; set lsmeandt;   if str1='control' then call     symput('plsm', lsmean);   if str1='str1' then call     symput('tlsm', lsmean); *-----* </pre>	<p>What does PROC GLM have to do with the median?</p>
$\bar{Y}_a - \bar{Y}_b \pm t_{1-\alpha/2, df} \cdot S(\bar{y}_a - \bar{y}_b) \quad (1)$	<p>Want something quick. Don't want to deal with formulas.</p>	

I should have read a little further. Two pages later, the code for the solution was there:

### ODS OUTPUT QUANTILES; PROC UNIVARIATE CIPCTLDF ...

Figure 15 (Guangbin, 2003)

<p>asymmetric values of <math>l</math> and <math>u</math>. PROC UNIVARIATE in SAS version 8 provides a convenient way to obtain the distribution-free confidence interval for the quartiles. The following code shows how to get a 95% distribution-free confidence interval for the median.</p>	<pre> proc univariate data=final loccount   modes cibasic(alpha=.05)   cipctldf(TYPE=ASYMMETRIC   alpha=.05);   var P&amp;var ;   by &amp;strt ; run;  data pct; set pctldf;   if quantile='50% Median' ; run;  ods output close; ods listing ; </pre>
<pre> *-----* * The following statements get the 95% * * confidence interval for each * * treatment group based on order * * statistics * *-----* ods listing close; ods output quantiles=pctldf ; </pre>	

## FOLLOW-UP: MORE TARGETED SEARCH

Instead of reading on, however, I tried another search. It turns out that I needed to include the words PROC UNIVARIATE. In other words, I had to guess that if PROC UNIVARIATE computed the confidence interval for the mean, perhaps it also had an option to do so for the median. These results are shown in Figure 16:

Figure 16

Google search results for "sas confidence interval median proc univariate". The search bar contains the text "sas confidence interval median proc univariate" and a "Search" button. The results show two main entries:

- The UNIVARIATE Procedure: PROC UNIVARIATE Statement** - 5:53pm  
Feb 24, 2009 ... **confidence intervals** for the mean, standard deviation, and variance .... specifies the **SAS** catalog that **PROC UNIVARIATE** uses to save traditional graphics output. ... the **median** absolute deviation about the **median** (MAD), ...  
support.sas.com/.../procstat/.../procstat\_univariate\_sect008.htm -  
Cached - Similar - [Share] [Print] [Close]
- The UNIVARIATE Procedure: Robust Estimators**  
Feb 24, 2009 ... What's New in the Base **SAS** Statistical Procedures .... **PROC UNIVARIATE** provides several methods for robust estimation of location and scale. ... The Winsorized **confidence interval** for the location parameter has upper and lower limits ... A very robust scale estimator is the MAD, the **median** absolute ...  
support.sas.com/.../procstat/.../procstat\_univariate\_sect031.htm -  
Cached - Similar - [Share] [Print] [Close]

Below these results are links for "SAS code - University of Manitoba" and "Calculating a Nonparametric Estimate and Confidence Interval Using ...".

The first result, a link to the PROC UNIVARIATE Version 9.2 documentation, is particularly useful. It has all of the options for computing both normal-approximation and distribution-free confidence intervals median, as well as a sample program. There is a wealth of easy-to-use information here, a portion of which is shown in Figure 17:

Figure 17 (SAS Institute, 2010 [2])

The ODS SELECT statement restricts the output to the "Quantiles" table; see the section [ODS Table Names](#). The CIQUANTNORMAL option produces confidence limits for the quantiles. As noted in [Output 4.10.1](#), these limits assume that the data are normally distributed. You should check this assumption before using these confidence limits. See the section [Shapiro-Wilk Statistic](#) for information about the Shapiro-Wilk test for normality in PROC UNIVARIATE; see [Example 4.19](#) for an example that uses the test for normality.

```

title 'Analysis of Female Heights';
ods select Quantiles;
proc univariate data=Heights ciquantnormal(alpha=.1);
  var Height;
run;

```

Quantiles (Definition 5)			
Quantile	Estimate	90% Confidence Limits	
		Assuming Normality	
100% Max	70.0		
99%	70.0	68.94553	70.58228
95%	68.6	67.59184	68.89311
90%	67.5	66.85981	68.00273
75% Q3	66.0	65.60757	66.54262
50% Median	64.4	64.14564	64.98770

## INTERNATIONAL ENCODING METHODS

Companies often receive datasets created in different countries in using different encoding methods due to language differences. Encoding establishes the environment to process SAS syntax and to read and write SAS data sets. Encoding issues present difficult problems. First, without the right encoding, you cannot open the dataset. Secondly, these problems tend to be random. Therefore, the likelihood that someone has solved your particular problem is unfortunately low.

In this next example, I describe using a Google search to eventually open a data from a Chinese affiliate. Figure 18 shows the error message we got in trying to the read the dataset:

Figure 18

```

1      options nomtterr;
2      libname in 'C:\test_drug\china';
NOTE: Libref IN was successfully assigned as follows:
      Engine:          V9
      Physical Name: C:\test_drug\china
3
4      data eg;
5      set in.eg;
6      run;

NOTE: Format VISF was not found or could not be loaded.
NOTE: Format ECGRSLF was not found or could not be loaded.
NOTE: Format YNF was not found or could not be loaded.
ERROR: Some character data was lost during transcoding in the dataset IN.EG.
ERROR: Some character data was lost during transcoding in the dataset IN.EG.
ERROR: Some character data was lost during transcoding in the dataset IN.EG.
NOTE: The data step has been abnormally terminated.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: SAS set option DBS=0 and will continue to check statements. This may cause
NOTE: There were 226 observations read from the data set IN.EG.
WARNING: The data set WORK.EG may be incomplete. When this step was stopped ther
NOTE: DATA statement used (Total process time):
      real time          0.18 seconds
      cpu time           0.00 seconds

```

The Chinese-speaking programmers contacted the group that created the dataset, but still could not resolve the issue. I then tried to solve the problem. Unfortunately I know very little about this area of SAS.

**GOOGLE THE ERROR MESSAGE**

When you know nothing about the problem, sometimes Googling the error message itself works. I tried searching **SAS SOME CHARACTER DATA WAS LOST DURING TRANSCODING**. Unfortunately, the search returned a lot of technical manuals. These were aimed at IT administrators dealing with global deployments, and not at all helpful to me.

**BROADER FOLLOW-UP SEARCH**

By this point, I was running out options. I next tried searching **SAS CHINESE ENCODING**. This returned some useful information; however, none of it worked.

Figure 19 (SAS Institute, 2003)

**Windows Double-Byte Encodings**

PLATFORM	LANGUAGE	ENCODING=VALUE
Windows	Japanese	shift-jis or ms-932
Windows	Korean	euc-kr or ms-949
Windows	Simplified Chinese	euc-cn or ms-936
Windows	Traditional Chinese	big5 or ms-950

From the documentation, it looks like this should have worked

```

data eg;
set in.eg(encoding='ms-950');
run;

```

NOTE: Format VISF was not found or could not be loaded.  
NOTE: Format ECGRSLF was not found or could not be loaded.  
NOTE: Format YNF was not found or could not be loaded.  
ERROR: Some character data was lost during transcoding in the dataset  
ERROR: Some character data was lost during transcoding in the dataset  
ERROR: Some character data was lost during transcoding in the dataset  
NOTE: The data step has been abnormally terminated.  
NOTE: The SAS System stopped processing this step because of errors.

Suffice it to say after a lot of trial and error, I stumbled upon the solution. For some reason, using ASCIIANY as the encoding option opened the dataset.

Figure 20 (SAS Institute, 2010)

The screenshot shows the SAS documentation page for the ENCODING= Data Set Option. The page title is "SAS(R) 9.2 National Language Support (NLS): Reference Guide". The left sidebar contains a navigation menu with categories like "Data Set Options for NLS" and "ENCODING= Data Set Option". The main content area is titled "ENCODING= Data Set Option" and includes a "Syntax Description" section. A red box highlights the text "By trial and error" with an arrow pointing to the "ASCIIANY" option in the "Syntax Description" table. The table lists options: ANY, ASCIIANY, and EBCDICANY, each with a description of when transcoding occurs or does not occur. A note states: "Note: ANY is a synonym for binary. Because the data is binary, the actual encoding is irrelevant."

Why did ASCIIANY work, and the other options did not? The documentation in Figure 21 provides some of the answer, but honestly, I don't know, and I really don't care. I am glad to be done with this problem. It is unlikely to come up again in this particular form. Other encoding problems may well require different solutions. But this example is instructive in that shows how to use Google to solve weird SAS problems, even if you know very little about the issue.

Figure 21 (SAS Institute, 2010)

```
1      options nofmterr;
2      libname in 'C:\test_drug\china';
NOTE: Libref IN was successfully assigned as follows:
      Engine:          V9
      Physical Name: C:\test_drug\china
3
4      data eg;
5      set in.eg(encoding='asciiany');
6      run;
```

NOTE: Format VISF was not found or could not be loaded.  
NOTE: Format ECGRSLF was not found or could not be loaded.  
NOTE: Format YNF was not found or could not be loaded.  
NOTE: There were 1509 observations read from the data set IN.EG.  
NOTE: The data set WORK.EG has 1509 observations and 27 variables.

**ASCIIANY**  
Transcoding normally occurs when SAS detects that the session encoding and data set encoding are different. ASCIIANY enables you to create a data set that SAS will not transcode if the SAS session that accesses the data set has a session that encoding value of ASCII. If you transfer the data set to a machine that uses EBCDIC encoding, transcoding occurs.

## CONCLUSIONS

In summary, you can leverage the experience of others by using Google searches to solve SAS problems. As the examples in this paper show, you can solve new problems you and really expand your skills. Using Google searches effectively requires some knowledge of SAS, and the ability to recognize the solution once you have found it. In general, I look for solutions from a reliable source (particularly support.sas.com), that are clear, concise, easy to implement, and do not require a lot of follow-up validation. These tend to be solutions based on PROCs or functions.

## REFERENCES

- Chung, Chang Y and Dunn, Toby (2005), "Page X of Y with Proc Report", Paper CC31, Proceedings of the Pharmaceutical Industry SAS® Users Group Conference 2005, <http://www.lexjansen.com/pharmasug/2005/CodersCorner/.%5CCC31.pdf>
- Mason, Phil (2007), "SAS 9 Tips · Part II", MeasureIT, Issue 5.11, Computer Measurement Group, November, 2007, [http://www.cmg.org/measureit/issues/mit46/m\\_46\\_1.html](http://www.cmg.org/measureit/issues/mit46/m_46_1.html)
- Peng, Guangbin (2003), "Confidence Intervals in Analysis and Reporting of Clinical Trials", Proceedings of the Pharmaceutical Industry SAS® Users Group Conference 2003, <http://www.lexjansen.com/pharmasug/2003/statisticspharmacokinetics/sp050.pdf>
- SAS Institute, Inc. (2003), "TS-691: SAS® Encoding Values, IANA Preferred MIME Charset, Java™ and Oracle® Encoding Names", Knowledge Base, Papers, SAS Technical Papers, <http://support.sas.com/techsup/technote/ts691.pdf>
- SAS Institute Inc. (2004), "Sample 24808: Calculating Age with Only One Line of Code", Knowledge Base, Samples & SAS Notes, <http://support.sas.com/kb/24/808.html>
- SAS Institute Inc. (2006), "Usage Note 15727: Writing PAGE X OF Y in RTF does not work with BODYTITLE", Knowledge Base, Samples & SAS Notes, <http://support.sas.com/kb/15/727.html>
- SAS Institute, Inc. (2010), "ENCODING= Data Set Option", SAS(R) 9.2 National Language Support (NLS): Reference Guide, <http://support.sas.com/documentation/cdl/en/nlsref/61893/HTML/default/viewer.htm#/documentation/cdl/en/nlsref/61893/HTML/default/a002601944.htm>
- SAS Institute Inc. (2010)[2], "PROC UNIVARIATE: SAS: Example 4.10 Computing Confidence Limits for Quantiles and Percentiles", Base SAS(R) 9.2 Procedures Guide: Statistical Procedures, Third Edition, [http://support.sas.com/documentation/cdl/en/procstat/63104/HTML/default/viewer.htm#/documentation/cdl/en/procstat/63104/HTML/default/procstat\\_univariate\\_sect065.htm](http://support.sas.com/documentation/cdl/en/procstat/63104/HTML/default/viewer.htm#/documentation/cdl/en/procstat/63104/HTML/default/procstat_univariate_sect065.htm)
- SAS Institute Inc. (undated), "Pageof macros (.sas)", Knowledge Base, Focus Areas, Base SAS, ODS PDF, Archive: SAS 8.2 ODS PRINTER Family, [http://support.sas.com/rnd/base/ods/odsprinter/pageofpp\\_public.sas](http://support.sas.com/rnd/base/ods/odsprinter/pageofpp_public.sas)
- Schreier, Howard (2005), "Re: In proc SQL, how to do something similar to lag function in data step?", SAS-L posting, 30 August 2005, <http://listserv.uga.edu/cgi-bin/wa?A2=ind0508e&L=sas-l&D=0&P=18631>
- Tong, Cindy (2003), "ODS RTF: Practical Tips", Proceedings of the Northeast SAS® Users Group Conference, 2003, <http://www.nesug.org/proceedings/nesug03/at/at007.pdf>

## ACKNOWLEDGMENTS

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 registered trademarks or trademarks of their respective companies.

## CONTACT INFORMATION

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

Michael Todd  
Nth Analytics  
12 Crimson King Trail  
Flemington, NJ 08822

Work Phone: 908.672.5649  
Fax: 253.595.7413  
Email: info@nthanalytics.com  
Web: www.nthanalytics.com