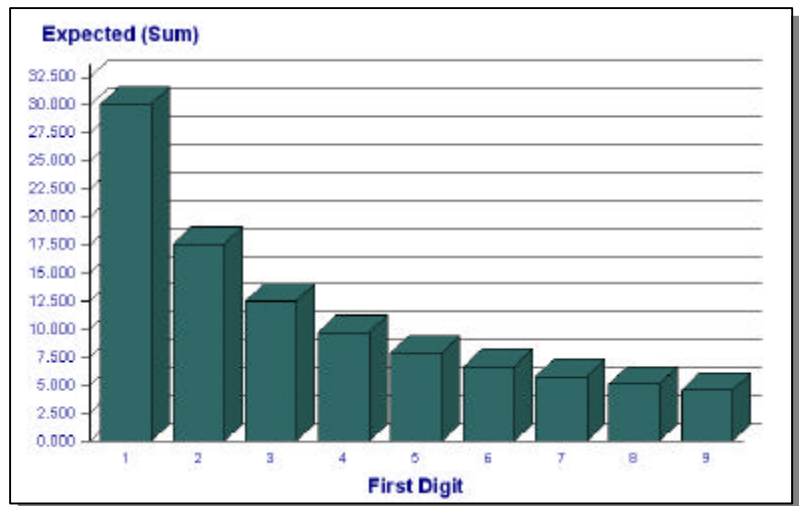


Detecting Anomalies in Your Data Using Benford's Law

Curtis A. Smith, Defense Contract Audit Agency, La Mirada, CA

ABSTRACT

Analyzing large amounts of data looking for anomalies can be a disheartening task. You need techniques that will allow you to quickly assess the data in ways that will highlight potential anomalies while keeping you from chasing the wind. Benford's Law is one such technique. Using Benford's Law and SAS® Software you can quickly identify one or more first digit patterns that defy statistical averages. Within this paper, the author will present SAS code that will enable you to quickly and easily find anomalies in the data you analyze. The SAS code will include the Data Step, the Merge statement, and the FREQ, REPORT, and GPLOT procedures. The author will also present some findings from the data he analyzes. The technique presented is powerful, yet easy to understand and use.



INTRODUCTION

Benford's law is so named after Dr. Frank Benford. Dr. Benford was a physicist working for General Electric in the 1930's. He noticed that certain pages of his logarithm book were more worn than others. After some study he realized that within a large enough universe of numbers that were naturally compiled that the first digits of the numbers would occur in a logarithmic pattern. The first digits of numbers are the non-zero, absolute value integers. For example, the first digit of 1 is 1; the first digit of 10 is 1; the first digit of -100 is 1; and the first-two digits of 1200 are 12. Dr. Benford tested 20,229 sets of numbers from many unrelated types of data. He found the same pattern to always exist. This statistical oddity provides an opportunity to those who need to analyze vast amounts of data for anomalies. If the first digits within the numbers of the data you are analyzing do not follow the Benford pattern, then something unnatural has happened with your data.

SO, WHAT'S THIS BENFORD'S LAW LOOK LIKE, ANYWAY?

Simply stated, any digit or combination of digits will follow the following logarithmic pattern:

$$x = \log_{10} \text{ of } 1 + (1/n)$$

Where n is the digit or combination of digits being tested and x is the percentage of their occurrence.

Benford's Law provides auditors with the expected digit frequencies in tabulated data. By examining the digit and the number frequencies, auditors can gain data insights that might be missed using traditional analytical procedures and sampling methods. The digit and number patterns could point to number invention, systematic frauds, data errors, or biases in the data.

- Dr. Mark Nigrini

(By the way, I created this graph with SAS/Graph's Graph-N-Go.) So, why is the first digit of 1 so much more prevalent than other digits? And why the declining distribution from 1 to 9? Perhaps Benford's Law can be explained this way. In a series of numbers, to go from 10 to 20 requires a 100% increase, but to go from 20 to 30 requires a 50% increase, and so forth. So, if numbers are being incremented, it takes less incrementing to go from 8 to 9 than it does to go from 1 to 2. Once the number increments from 8 to 9, then with only a little incrementing, the 9 will increment to 10 (a first digit of 1). So, numbers tend to fall within a first digit of 1 more than any other digit.

However, this distribution will not exist within every set of numbers. First, the universe of numbers must be large enough for the distribution to take shape. Some have found that a universe smaller than 100 items will not exhibit the pattern. Second, the numbers must be free of artificial limits or origins. For example, when evaluating a data file of travel claims you might find that the first-two digit combination of 24 exists greater than expected with Benford's Law. This might happen if the company has a policy that reimbursement claims for \$25 and above must be supported with receipts - so travelers claim a lesser amount, such as \$24.95. If you analyze the transactions in my checkbook you will find the first-four digit combination of 1995 to occur at a high rate. This is because my ISP always charges me a monthly rate of \$19.95.

HOW CAN BENFORD'S LAW HELP YOU?

If you are in the business of analyzing data, such as the noble profession of auditing, you might need to look for areas of fraud or areas of oddities. Dr. Mark J. Nigrini and others have successfully used Benford's Law to detect potential fraud. Dr. Nigrini termed the use of analyzing digits within numbers as "digital analysis." It is difficult for the fraudster to avoid detection from digital analyses because the fraudster typically cannot influence an entire data file. Thus, the fraudster will invariably alter numbers in such a way that destroys the Benford distribution. But, you don't have to be looking for fraud to benefit from Benford's Law. There are many non-fraudulent reasons

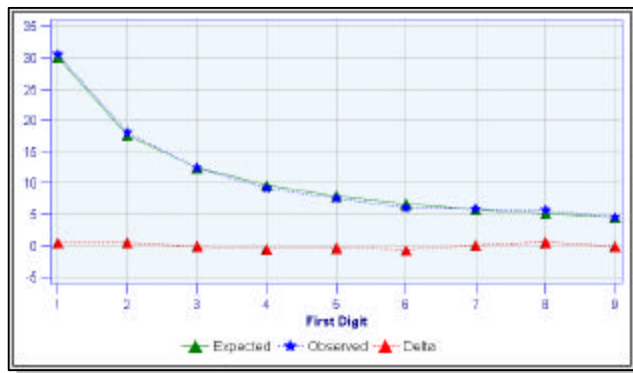
Considering the first digits 1 to 9, the expected distribution is:

why a universe of numbers can violate Benford's Law, yet still warrant your investigation.

Start Digging

Here are some samples from labor transaction data I have analyzed. First, look at a first digit analysis. Notice the table showing the observed versus expected distribution and delta, then notice the plot showing the same.

First Digit(s)	Observed Frequency Count	Observed Frequency Percent	Expected Frequency Percent	Observed - Expected Frequency Percent
1	419	30.562	30.103	0.459
2	248	18.089	17.609	0.480
3	170	12.400	12.494	-0.094
4	126	9.190	9.691	-0.501
5	104	7.586	7.918	-0.332
6	83	6.054	6.695	-0.641
7	81	5.908	5.799	0.109
8	78	5.689	5.115	0.574
9	62	4.522	4.576	-0.054
	1,371	100.000	100.000	-0.000



Nice curves. In this case, everything looks as Benford predicted. So, a dead end, right? No, just an opportunity to dig further.

Dig Deeper Using Multiple Digits

You can analyze the data using a combination of digits, such as the first-two digits. This can help find anomalies not apparent in a first digit analysis and can isolate further the anomalies found in a first digit analysis. Let's look at

First Digit(s)	Observed Frequency Count	Observed Frequency Percent	Expected Frequency Percent	Observed - Expected Frequency Percent
10	67	4.887	4.139	0.748
11	60	4.376	3.779	0.596
12	43	3.136	3.476	-0.340
13	39	2.845	3.218	-0.374
14	43	3.136	2.996	0.140
15	44	3.209	2.803	0.406
16	36	2.626	2.633	-0.007
17	30	2.188	2.482	-0.294
18	28	2.042	2.348	-0.306
19	29	2.115	2.228	-0.112
20	40	2.918	2.119	0.799

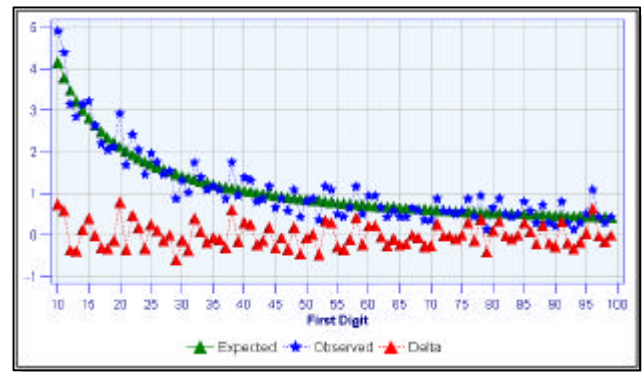
an example. First, take a look at a portion of the table report.

Benford's Law gives auditors the expected frequencies of the digits in tabulated data. The premise is that we would expect authentic and unmanipulated data to exhibit these patterns. If a data set does not follow these patterns, however, a few possible reasons exist to explain this phenomenon:

1. The data set did not meet the three tests, and/or,
2. The data set includes invented numbers, biased numbers, or errors.

- Dr. Mark Nigrini

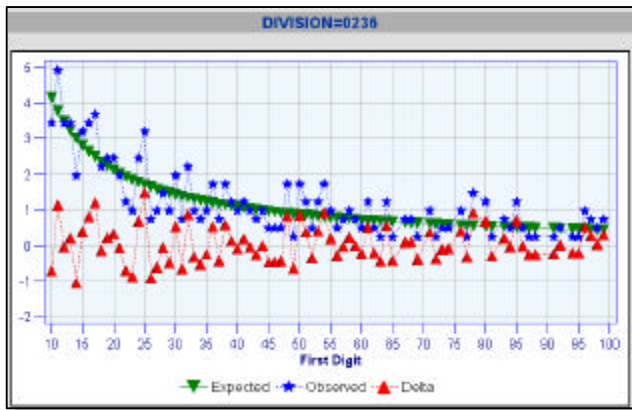
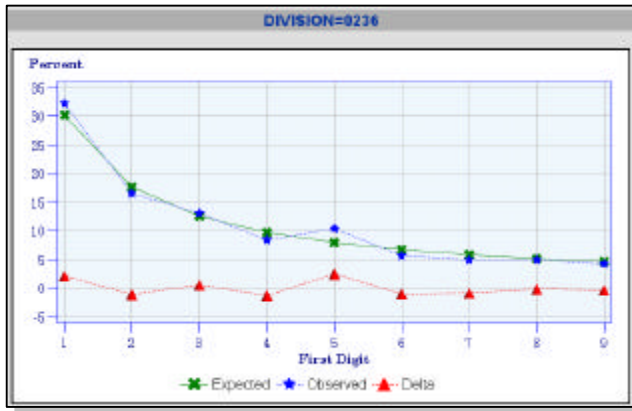
Then, look at the plot of the same information.



Notice here the anomalies beginning to show themselves. The X (green) line represents the Benford expected distribution - a very nice curve, indeed. The star (blue) line is the observed, and the triangle (red) the delta. You can quickly see the anomalies. Yet, in this example, the anomalies are not that great.

Dig Deeper By Subsetting

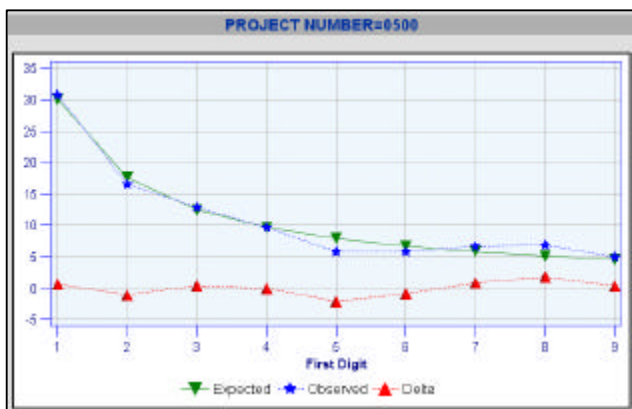
Another technique you can use is to subset your data. For example, if you analyze all of the labor transactions together by first digit, first-two digits, and so forth and find nothing, you might then look at subsets of the data. Rather than looking at all the labor transactions together, look at subsets by department, or by week, or by employee. You might then see anomalies showing up. Why? Because if one department or employee is doing something funny, or if something funny was being done during one week, looking at the entire universe can obscure the funny business. But isolating subsets can be revealing. Here are some examples of first digit and first-two digits analyses of labor data subset by division and then by project.



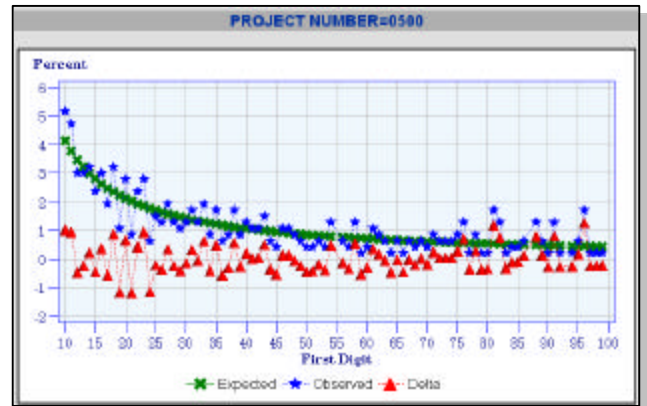
Notice in the first digit analysis for this one division there is an anomaly in the first digit of "5". While not a huge variance from the expected, a variance nonetheless. Then, looking at the same division at the first-two digits you can see anomalies all over the place. There are significant variances at values "50" and "54", both of which have a first digit of "5". But notice the even more significant variance at the value "25". This variance didn't even register in our first digit analysis. This is evidence of why going deeper than just the first digit analysis can be useful.

Let's look even further, by subsetting our universe by project.

Here is one of the projects in the universe as seen by a first digit analysis. There are small anomalies at the first digit of "5" and "8".

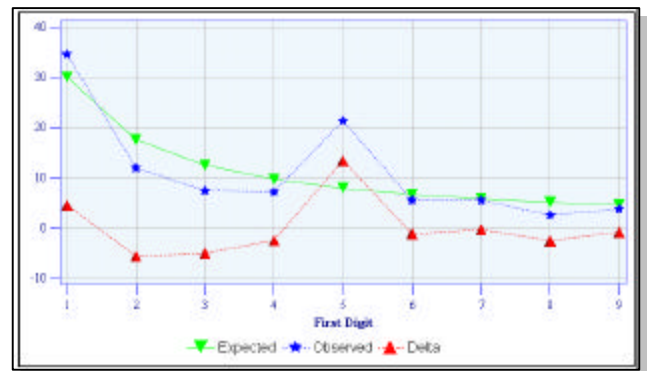


Looking at the same project using a first-two digits analysis you can see anomalies all over the place.

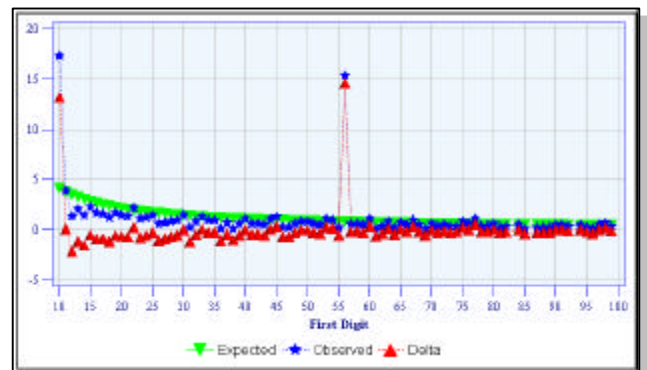


Big Findings

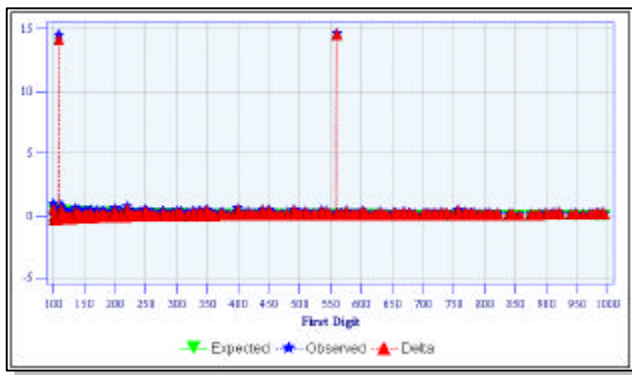
Well, enough of this fooling around. Let's take a look at some striking examples. Subsetting the data by location turned out to be much more revealing. First, check out the first digit analysis.



Here you can see anomalies all over the place - but the anomaly at first digit "5" really gets our attention. So, let's dig a little deeper. Feast your eyes on the first-two digit analysis.



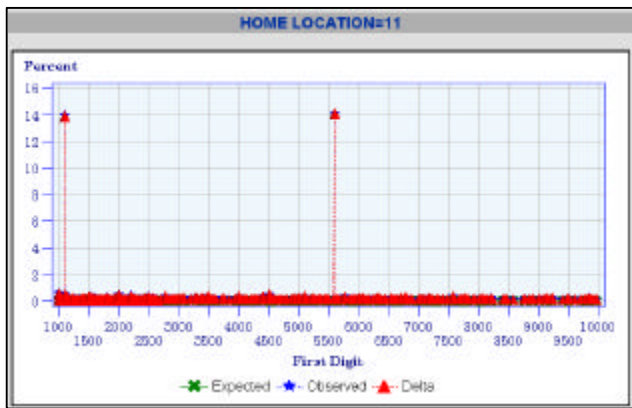
Wow! That first digit "5" anomaly turns out to be a really big first-two digit "56" anomaly. And, look at that first-two digit "10" anomaly that was hardly noticeable in the first digit analysis. This is going so well, let's dig even deeper. Take a gander at a first-three digit analysis.



Now we can pinpoint the two great anomalies at "109" and "560". We can easily determine these values and their frequency by referring back to the table report, a portion of which is shown below.

HOME LOCATION	First Digit(s)	Observed Frequency Count	Observed Frequency Percent	Expected Frequency Percent	Observed - Expected Frequency Percent
11	560	120	14.201	0.077	14.124
	109	119	14.083	0.397	13.686
	100	8	0.947	0.432	0.515
	113	7	0.828	0.363	0.446
	220	7	0.828	0.197	0.631

You might be wondering if you could go any deeper. Certainly. Let's take a look at a first-four digit analysis.



In this case, the first-four digit analysis did not add any value because the two anomalies are at "1090" and "5600". Because the numeric variable is dollars, the "1090" value is probably \$10.90 or \$109.00 and the "5600" value is probably \$56.00, \$560.00, or \$5,600.00. So, in both cases, digging to the next first digit will just add another zero to the end of our number, which won't help pinpoint the anomalies any better.

What to Do

So, what can you do with these anomalies? You can start to cross check them, looking for something in common. Considering our example, look to see if the anomalies within divisions also occur within projects. Then, you can begin extracting the actual data records that contain the anomalies and use the information on those data records to get to the root of the anomalies.

SO, HOW WAS SAS USED?

To produce these results, I used SAS to do five main tasks, which are as follows:

- ❑ Determine the observed distributions of the first digits
- ❑ Determine the expected distributions of the first digits
- ❑ Merge the observed and expected distributions and compute the deltas
- ❑ Create a report of the observed versus expected distribution
- ❑ Create a plot of the observed versus expected distribution

That doesn't seem too difficult. Let's look at the code at a high level. The code shown is for a first digit analysis for an entire universe. The modifications to do a first-two digits or more digit combination and to do a BY group analysis are not too different and will not be presented.

A bank auditor found that credit card balances written off as uncollectible had an excessive level of numbers with first-two digits 49. The investigation found that \$5,000 was an internal write-off limit for internal collections employees. One employee was responsible for most of the 49s by working with friends and having them apply for a card and then running up a balance to just below \$5,000. The employee would then write the debt off. The systematic nature of the fraud was evident from the first-two digits graph.

- Dr. Mark Nigrini

Determine the Observed Distributions of the First Digits

Here's the fundamental code I use to determine the distributions of the first digits.

```
DATA WORK.OBSERVED
  (KEEP=FIRSTDGT COUNT VAR INDEX=(FIRSTDGT));
  SET IN.INFILE.;
  FIRSTDGT=
    INPUT(SUBSTR(SCAN(PUT(VAR, BEST8.), 1), 1, 1),
      BEST8.);
  COUNT=1;
RUN;
PROC FREQ DATA=WORK.OBSERVED;
  TABLES FIRSTDGT/OUT=WORK.BENFORD
    (RENAME=(PERCENT=OBSERVED));
RUN;
```

In the code above, the user-specified file to analyze (IN.INFILE) is read and a new variable, FIRSTDGT, is created. This new variable is created by using the PUT function to convert the user specified numeric variable (VAR) to a character string. Then the SCAN function gets the first digit from the converted value. Then the INPUT function stores that first digit to the new numeric variable. Another new variable, COUNT, is created and set to 1. This will be used to summarize the frequency of the first digit.

Next, the FREQ procedure is used to create the frequency distribution of the first digit.

Determine the Expected Distributions of the First Digits

Here's the fundamental code to create the expected distributions.

```
DATA WORK.EXPECTED(INDEX=(FIRSTDGT) DROP=I);
  FORMAT EXPECTED 8.3;
  DO I = 1 TO 9;
    FIRSTDGT=I;
    EXPECTED=(LOG10(1+(1/I))*100);
    OUTPUT;
  END;
```

This data step simply creates a new data set that contains the first digit and the result of the log10 formula demonstrated by Dr. Benford.

Merge the Observed and Expected Distributions and Compute the Deltas

Here's the basic code for creating a data set with the observed and expected first digit distributions and the deltas for each digit.

```
DATA OUT.BENFORD;
  MERGE WORK.EXPECTED(IN=A)
        WORK.BENFORD(IN=B);
  BY FIRSTDGT;
  IF B;
  DELTA=SUM(OBSERVED, -EXPECTED);
RUN;
```

This data step simply creates a new data set by merging the observed and expected data sets by the first digit and creates a new variable, DELTA, containing the difference between the observed and expected distributions.

Create a Report of the Observed Versus Expected Distribution

Here's the fundamental code to create the tabular report.

```
PROC REPORT DATA=OUT.BENFORD NOWINDOWS
  HEADSKIP MISSING;
  COL FIRSTDGT COUNT OBSERVED EXPECTED DELTA;
  DEFINE FIRSTDGT /ORDER 'FIRST DIGIT(S)'
    WIDTH=5;
  DEFINE COUNT /'OBSERVED FREQUENCY COUNT'
    WIDTH=12 FORMAT=COMMA12.;
  DEFINE EXPECTED /'EXPECTED FREQUENCY
  PERCENT' WIDTH=12 FORMAT=8.3;
  DEFINE OBSERVED /'OBSERVED FREQUENCY
  PERCENT' WIDTH=12 FORMAT=8.3;
  DEFINE DELTA /'OBSERVED - EXPECTED FREQUENCY
  PERCENT' WIDTH=12 FORMAT=8.3;
  RBREAK AFTER /SUMMARIZE OL UL;
RUN;
```

This code is simply a REPORT procedure with the key variables defined.

Create a Plot of the Observed Versus Expected Distribution

Here's the fundamental code to create the overlay plot.

```
PROC GPLOT DATA=OUT.BENFORD;
  PLOT EXPECTED*FIRSTDGT OBSERVED*FIRSTDGT
  DELTA*FIRSTDGT / OVERLAY;
RUN;
```

This code simply uses the GPLOT procedure to create an overlay plot of the OBSERVED, EXPECTED, and DELTA variables over the first digit variable.

My actual code is a bit more complicated, as I use macro variables to allow the user to make specifications before running the application. I also use ODS statements to make HTML files of the REPORT and GPLOT procedure output. And I use ActiveX controls in my GPLOT procedure output.

CONCLUSION

Analyzing large amounts of data for anomalies or potential fraud does not have to be a disheartening task. Using digital analyses, such as Benford's Law, you can easily find anomalies in your data. It was not my intention within this paper to provide all of the SAS code I used to create the output shown within this paper. If you would like my code, send me an e-mail request. For a limited time my code is free, in exchange for your Benford's Law success stories.

REFERENCES

"Following Benford's Law, or Looking Out for No. 1",
Malcolm W. Browne,
<http://courses.nus.edu.sg/course/mathelmr/080498sci-benford.htm>

"The Power of One", Robert Matthews,
<http://www.newscientist.com/ns/19990710/thepowerof.html>

"Digital Analysis: a Computer-Assisted Data Analysis
Technology for Internal Auditors", Mark J. Nigrini, Ph.D.,
<http://www.itaudit.org/forum/emergingissues/f108ei.htm>

ACKNOWLEDGMENTS

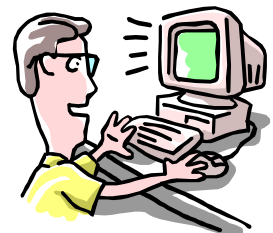
SAS is registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

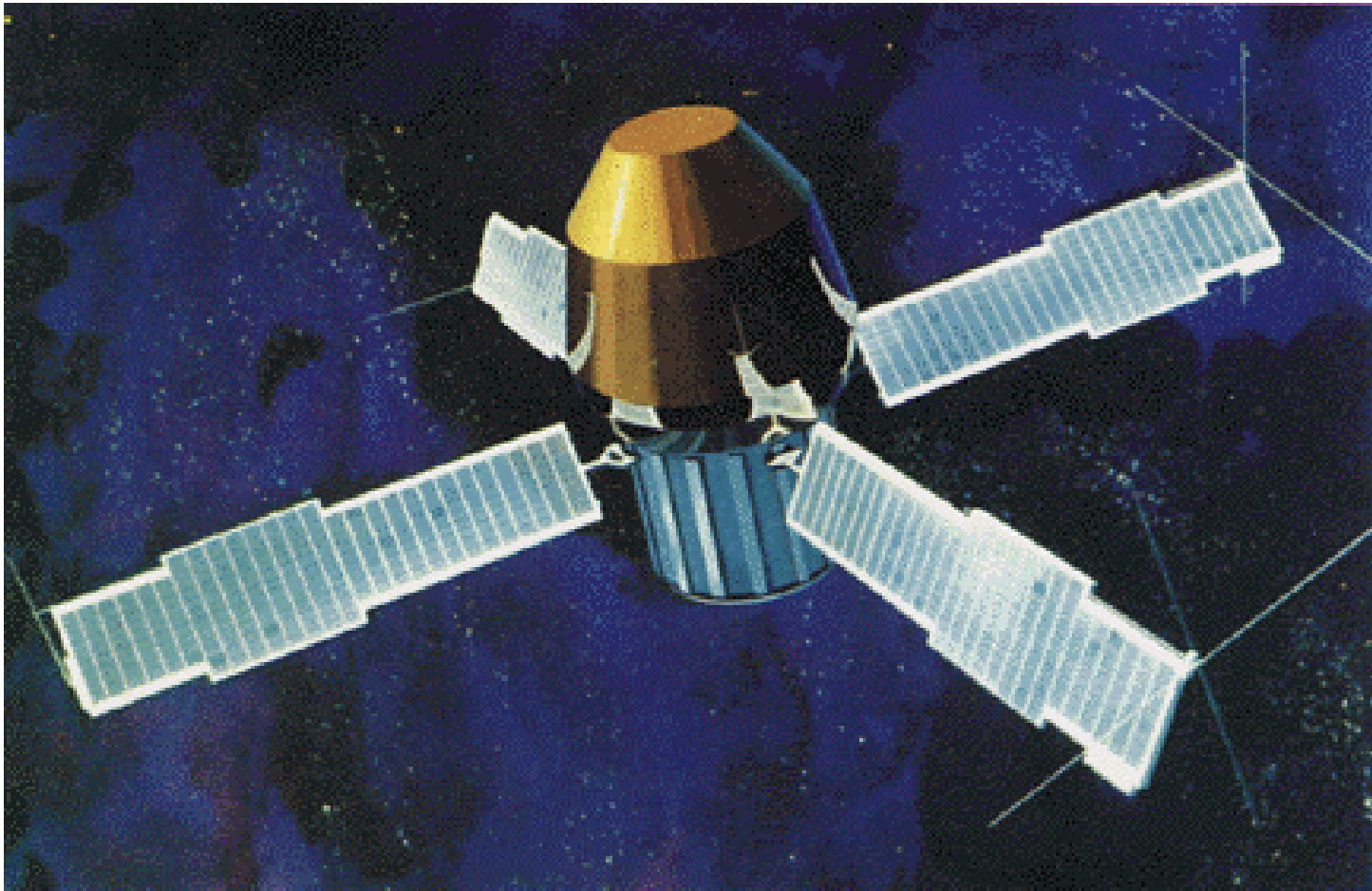
Your comments and questions are valued and encouraged. Contact the author at:

Curtis A. Smith
Defense Contract Audit Agency
P.O. Box 20044
Fountain Valley, CA 92728-0044
Work Phone: 714-896-4277
Fax: 413-383-6395
email: casmith@mindspring.com



Infinite Space

- SAS 2: Small Astronomical Satellite tells us our universe is expanding



Finite Space

An 8 1/2" x 11" piece of paper

An 8 1/2" x 11" piece of paper has a limit

An 8 1/2" x 11" piece of paper has a limit to the amount of data

An 8 1/2" x 11" piece of paper has a limit to the amount of data that it can display.

An 8 1/2" x 11" piece of paper has a limit to the amount of data that it can display and still be readable.

-
-
-

More **INFORMATION** Than Column Width

Squeezing **INFORMATION** into a narrow column can be easy if you are working with PROC REPORT - Simply turn on the FLOW option.

-
-
-

More INFORMATION- Than Column Width

Wouldn't it be nice if
DATA _NULL_
had a similar option?

Now you can use the WORDWRAP macro . . .

•
•
•

The Word Wrap Macro

- Why should I use this macro?
 - Simple to use
 - Easy to maintain
 - Parameter checking
 - Flexible and expandable
 - Handles multiple text strings

•
•
•

Simple to use

- Only one parameter is needed - the variable to wrap
- Starting column defaults to 1
- Ending column defaults to `LINESIZE`

•
•
•

Easy to Maintain

- Is your data subject to change?
- Expected or unexpected increases to field length?
- Can new text strings be added at a moment's notice?
- If 'YES' then *%wordwrap!*

•
•
•

Parameter Checking

- Simple and quick checks are performed to verify parameters are passed in the correct form.
- End columns greater than Start columns are automatically swapped complete with log warning.

•
•
•

Flexible and Expandable

- Can be called before, in between, or after other PUT statements.
- Can easily append prefixes
 - COMMENT: text_string_here
- Can easily append suffixes
 - text_string_here - CONTINUED

•
•
•

Handles Multiple Text Strings

- `Wrapvars=comment1 comment2;`
- Simply add new text strings to the `wrapvars` parameter when needed!
- `Wrapvars=comment1 comment2 comment3;`

•
•
•

Parameters Defined

- Only one parameter required:

<u>PARAMETER</u>	<u>DESCRIPTION</u>	
wrapvars=	Text string(s)	(required)
strt_col =1	Start position	(optional)
end_col =	End position	(optional)
end =y	Go to next line?	(optional)
offset =0	Indentation	(optional)

Example 1

- TEST DATA with macro call using default variables
- Note: Beginning of each single word is uppercased.
- **data** test;
- comment1 = '12345678901234567890 This Is Test Data';
- comment2 = 'Backacheheadachefootpain Verylong Word String';
- comment3 = ' Leading Spaces In Data Between Words';
- **run;**
- **data _null_;**
- **set** test;
- **file print** col=col;
- * **Call wordwrap using defaults;**
- %**wordwrap**(wrapvars=comment1 comment2 comment3)
- **run;**

Example 1 Output

- Called using defaults:



Example 2

- TEST DATA with macro call using default variables
- Note: Beginning of each single word is uppercased.
- **data** test;
- comment1 = '12345678901234567890 This Is Test Data';
- comment2 = 'Backacheheadachefootpain Verylong Word String';
- **run;**
- **data _null_;**
- **set** test;
- **file print col=col;**
- **PUT @1 'Left Column' @;**
- **PUT @61 'Right Column' @;**
- %**wordwrap**(wrapvars=comment1 comment2, strt_col=21, end_col=40)
- **run;**

Example 2 Output

- Called after all PUT statements:
- **PUT @1** 'Left Column' @;
- **PUT @61** 'Right Column' @;
- **%wordwrap**(wrapvars=comment1 comment2, strt_col=21, end_col=40)

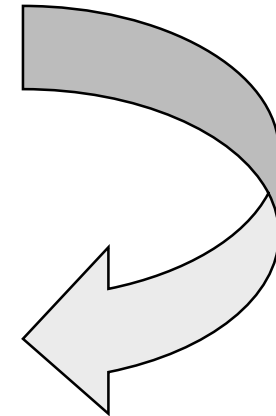
Column 21

Column 40

Left Column **12345678901234567890** Right Column

1) Backache... word is
"broken" or split as it is
longer than 20 characters.

This Is Test Data
Backacheheadachefoot
pain Verylong Word
String



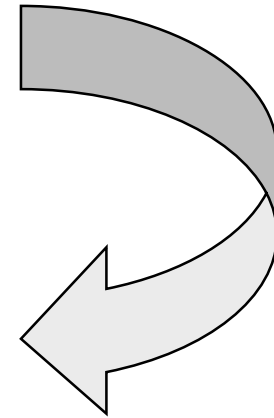
Example 3

- Called after all PUT statements:
- **PUT @1** 'Left Column' @;
- **PUT @61** 'Right Column' @;
- **%wordwrap**(wrapvars=comment2, strt_col=21, end_col=24)

Left Column **Back** Right Column

ache
head
ache
foot
pain
Very
long
Word

1) Only one text string is
passed in this example



Code Segment - Error Checking

- `%if %length(&end) gt 0 %then %let end = %upcase(%substr(&end,1,1));`
- `%let linesize=%sysfunc(getoption(linesize));`
- `%if %length(&end_col) eq 0 or &end_col gt &linesize %then %let end_col=&linesize;`
- `%let end_col=%eval(&end_col);`
- `%if %verify(&strt_col,0123456789) eq 0 %then %do;`
- `%let strt_col=%eval(&strt_col);`
- `%if &strt_col gt &end_col %then %do;`
- `%let swapit=&strt_col;`
- `%let strt_col=&end_col;`
- `%let end_col=&swapit;`
- `%put WORDWRAP MACRO WARNING: STRT_COL GREATER THAN END_COL, VALUES HAVE BEEN SWAPPED.;`
- `%end;`
- `%end;`

Code Segment - Implicit Arrays

- `%if %length(&_control) eq 0 %then %let _control=0;`
- `%if &_control ge 9999 %then %let _control=0;`
- `%let _control=%eval(&_control+1);`
- `* length of individual word, not string;`
- `length _word $200;`
- `array _cmt&_control &wrapvars;`
- `put @&strt_col @;`
- `do over _cmt&_control;`
- `_i=1;`
- `_word=scan(_cmt&_control,_i, ' ');`
- `...`
- `end;`

Code Segment - Wordwrap Loop

```
• do while (_word ne ' ');
•   _len=length(_word);
•   _width=&end_col-&strt_col+1;
•   if _len gt _width then do;
•     _numcuts=ceil(_len/_width);
•     do cntr = 1 to _numcuts;
•       cutword=substr(_word, _width*(cntr-1)+1, _width);
•       if (_len+&col_var-1) gt &end_col then put /
•       @(&strt_col+&offset) @;
•       put cutword @;
•     end;
•   end;
•   else do;
•     if ((_len+&col_var-1) gt &end_col) then put /
•     @(&strt_col+&offset) @;
•     put _word @;
•   end;
•   _i+1;
•   _word=scan(_cmt&_control,_i, ' ');
• end;
• end;
```

Example 4

- `pt_name = trim(pt_name)||'/';`
- `aeterm = upcase(aeterm);`
- `put @&c3 ae4flag $char4. @;`
- `put @&c5 ontrtflg $char7.-c @;`
- `put @&c6 daytoend 4. @;`
- `put @&c7 aestdt date7. @;`
- `put @&c8 dayontrt 3. @;`
- `put @&c9 duration 4. @;`
- `put @&c10 aecausal no_yess5.-c @;`
- `put @&c11 aesev newaesev6.-c @;`
- `put @&c12 aeacttrt newacttr. @;`
- `put @&c13 aecontrt @;`
- `put @&c14 aeactoth @;`
- `put @&c15 aeout newoutcm. @;`
- `%wordwrap(wrapvars=pt_name aeterm, strt_col=&c4,`
- `end_col=&c4+&wd4-1, end=y);`

-
-
-

Example 4 Output

•	_____ COLUMN 4 _____												
•	0063/10208 46/F/W/114 #	Nasal congestion/ INCREASED	PRE	-13	08MAR01 -15	3	No	Mod.	N/A	Yes	No		
•		STUFFY NOSE											
•	#	Diarrhoea NOS/ DIARRHEA	PRE	-11	10MAR01 -13	3	No	Mod.	N/A	No	No		
•		Nasopharyngitis/ COLD SX	PRE/ON	5	18MAR01 -5	10	No	Mod.	N/A	Yes	No		



-
-
-
-
-
-
-
-

Example 5

- data _null_;
- set teae (in=a) freq(in=b) end=_eof;
- retain pgflag pt_flag soc_flag;
- file print header=header ll=ll col=col;
- **contd = '- CONTINUED';**
- pos=&c1;
- ...
- if pt_flag and soc_flag then
- do;
- soc_name = upcase(trim(soc_name));
- %wordwrap(wrapvars=soc_name **contd**, strt_col=pos, end_col=&c1+&wd1-1);
- pt_flag = 0;
- end;
- pos=&c1+2;

Example 5 Output

MUSCULOSKELETAL, CONNECTIVE TISSUE
AND BONE DISORDERS

Arthralgia

JOINT ACHES

S5/1014 Placebo PRE No Mild N/A Yes No

Note: Wordwrap macro used with page flags and "Continued" variable.... Next page shown below:

~~~~~ NEW PAGE ~~~~~

15-MSL-234/101

Microsoft Labs

101/wtest\_01.lst 15AUG2001 12:15 Page 7 of 10

Table 15.1.2.3 Adverse Events by Body System, Preferred Term, and Reported Term (includes pre-, on-, and post-treatment)

| BODY SYSTEM/<br>Preferred term/<br>REPORTED TERM | Inv./<br>patient<br>number | Treatment<br>group<br>(µg/day) | Period<br>[a] | Possible<br>relation-<br>ship to<br>study med | Intensity<br>Study<br>med<br>[b] | Concom<br>med<br>[c] | Other<br>[d] |
|--------------------------------------------------|----------------------------|--------------------------------|---------------|-----------------------------------------------|----------------------------------|----------------------|--------------|
|--------------------------------------------------|----------------------------|--------------------------------|---------------|-----------------------------------------------|----------------------------------|----------------------|--------------|

MUSCULOSKELETAL, CONNECTIVE TISSUE  
AND BONE DISORDERS - CONTINUED

Arthralgia - continued

KNEE PAIN

S6/1019 None ON No Mod. None Yes No

KNEE PAIN (L)

S3/1013 Placebo ON No Mod. None Yes No

•  
•  
•

# Why Use This Macro?

- Why do I use this macro?
  - Simple to use
  - Easy to maintain
  - Parameter checking
  - Platform independent
  - Flexible and Expandable
  - Handles multiple text strings
  - Does not cut words unless it is necessary