Paper RF-006

Captain's LOG: Taking Command of SAS® Logarithm Functions

Britney D. Gilbert, Juniper Tree Consulting, Porter, OK
Joshua M. Horstman, Nested Loop Consulting, Indianapolis, IN

ABSTRACT

In BASE SAS®, there are multiple logarithmic functions available. The most used log functions are the natural and common log functions. However, the syntax of the natural log function can be easily confused with the mathematical notation for the common log and can lead to incorrect results. This intent of this paper is to explore the definitions of each of these log functions, to seek out full understanding, and to boldly go where no programmer has gone before.

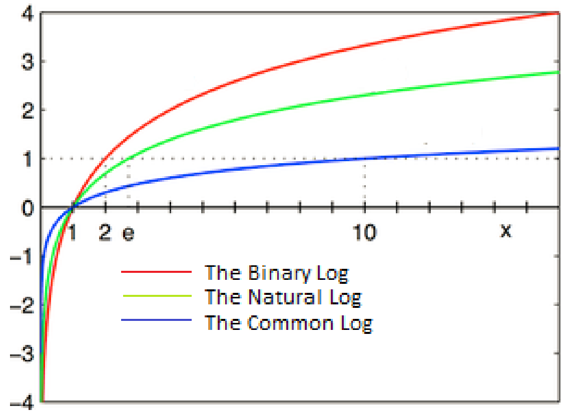
INTRODUCTION

The logarithm of a number is the exponent to which a base must be raised to produce that number. For example, the logarithm of 1000 to base 10 is the exponent 3, because 10 to the power 3 is 1000: 1000 = 10 × 10 × 10 = 10^3.[1]

Each discipline has their favorite logarithm bases that are prominent in their theories and algorithms. In computer science, it's the binary log, or logarithm base 2. For Physicists and Engineers, many of their applications use the common log, or logarithm base 10. And, for Mathematicians, they use the natural log, or logarithm base *e* (Euler's number ≈ 2.71828).

The confusion arises in the notations because there are overlapping logarithm shorthands with different meanings. Specifically, Engineers will denote the natural logarithm as y = log x while a Mathematician interprets this as log base 10 and denotes the natural log as y = ln x.

Figure 1. Graph of the Binary Log, Natural Log, and Common Log[2]



To demonstrate the different SAS LOG functions, This paper will use the most basic identities, trivial identities:

$\text{LOG}_b(1)$ = 0 because b^0 = 1

$\text{LOG}_b(b)$ = 1 because b^1 = b

EXAMINING THE SYNTAX OF LOG FUNCTIONS

In SAS, there are several logarithm functions available to the programmer. It is important to know the definition of the function that you are programming to ensure proper use. As discussed earlier, Mathematicians and Engineers denote the logarithm bases differently. So, do not assume the functional syntax is your discipline's shorthand.

THE NATURAL LOG

In SAS, the function for the natural log is LOG. In Example 1, the trivial identities are tested in a DATA step. The input of 1, returns the expected value of 0 and using a Euler's Number approximation of 2.71828, the result y is very close to 1.

In this case, we can use another SAS function to help demonstrate the LOG function. The CONSTANT function will return the value of a mathematical constant. For Euler's Number, the proper notation is 'E'. From reviewing the log using CONSTANT('E'), the expected value of 1 is returned.

Example 1. Natural log function: LOG

```
Data _NULL_;
 y = LOG(1);              put y "= LOG(1)";
 y = LOG(2.71828);        put y "= LOG(2.71828)";
 y = LOG(CONSTANT('E'));  put y "= LOG(e)";
Run;

0 = LOG(1)
0.9999993273 = LOG(2.71828)
1 = LOG(e)
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

To prevent more confusion, it is important to note that 'EULER' is also a valid input for the CONSTANT function, but this represents a different mathematical constant – *Euler's Constant* or 0.5772.

THE COMMON LOG

For the common log, the SAS function is LOG10. In Example 2, the trivial identities are again programmed in a DATA step. After reviewing the log we see the expected results of 0 and 1 are returned for the inputs of 1 and 10, respectively.

Example 2. Common logarithm function: LOG10

```
Data _NULL_;
 y = LOG10(1);  put y "= LOG10(1)";
 y = LOG10(10); put y "= LOG10(10)";
Run;

0 = LOG10(1)
1 = LOG10(10)
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

OTHER LOG FUNCTIONS

Addition logarithmic functions are available in SAS beyond the natural log and common log. Among these are the LOG2 function that returns the binary logarithm, or base 2, and the LOG1PX function that returns the logarithm of one plus the argument. LOG1PX is useful when the argument is close to zero, because it is more accurate than using LOG(1+x). The juxtaposition of LOG1PX and LOG(1+x) are beyond the scope of this paper, but more information can be found in *Base SAS® Language Reference*.

THE EXPONENTIAL FUNCTION AND OPERATOR

Quality checks in programming are always a good idea. One way to check to make sure that the appropriate log function was used is to check the results the inverse operation using exponents. An exponential (or antilog) with the same base as the log function will undo the operation.

EXPONENTIAL FUNCTION

The EXP function raises the input argument to the power of Euler's Number, e. This can be used to check or undo the natural log function LOG.

In Example 3, the EXP function has been used to undo LOG(1), LOG(2.71828), and LOG(CONSTANT('E')). Reviewing the log, each quality check applied has returned the original argument of the LOG function.

Example 3. Quality Checks Using the EXP Function

```
*************************************************;
*** Quality Checks for Natural Log Functions ***;
*************************************************;
Data _NULL_;
 y = LOG(1);                put y "= LOG(1)";
 x = EXP(y);                put x "= EXP(y)";

 y = LOG(2.71828);          put y "= LOG(2.71828)";
 x = EXP(y);                put x "= EXP(y)";

 y = LOG(CONSTANT('E'));    put y "= LOG(e)";
 x = EXP(y);                put x "= EXP(y)";
Run;

0 = LOG(1)
1 = EXP(y)
0.9999993273 = LOG(2.71828)
2.71828 = EXP(y)
1 = LOG(e)
2.7182818285 = EXP(y)
NOTE: DATA statement used (Total process time):
      real time           0.16 seconds
      cpu time            0.00 seconds
```

EXPONENTIAL OPERATOR

Although the exponential function can only be used for the LOG function, the exponential operator can be used to check any logarithm base. The exponential operator is a double asterisk (**) and can be used to invert logarithmic functions by raising the base of the logarithm to the power of the result of the logarithm. In Output 1, the code below was run using ten test values (TEST_VALUE):

```
_LOG_ = LOG(TEST_VALUE);
ANTILOG = CONSTANT('E')**_LOG_;

_LOG10_ = LOG10(TEST_VALUE);
ANTILOG10 = 10**_LOG10_;

_LOG2_ = LOG2(TEST_VALUE);
ANTILOG2 = 2**_LOG2_;
```

The results show that in each pairing the ANTILOG variables match the original test value. This is a good quality check to ensure that you are using the proper base in the logarithm function.

Output 1. Quality Checks Using Exponential Operators

	TEST_VALUE	LOG	ANTILOG	LOG10	ANTILOG10	LOG2	ANTILOG2
1	3623796	15.103032653	3623796	6.5591637413	3623796	21.789070311	3623796
2	6934123	15.751965144	6934123	6.8409915411	6934123	22.725281997	6934123
3	9658121	16.083309674	9658121	6.9848926421	9658121	23.203311107	9658121
4	2142582	14.577522202	2142582	6.330937452	2142582	21.030918989	2142582
5	469490	13.059402278	469490	5.6716263464	469490	18.840734904	469490
6	507302	13.136861766	507302	5.7052665745	507302	18.952485323	507302
7	4004022	15.202809914	4004022	6.602496455	4004022	21.93301847	4004022
8	9469009	16.063534813	9469009	6.9763045293	9469009	23.174782014	9469009
9	7016290	15.763745146	7016290	6.8461075312	7016290	22.742276948	7016290
10	5254037	15.474507291	5254037	6.7204931269	5254037	22.32499493	5254037

AN APPLICATION OF LOG FUNCTIONS

Logarithms have their place in nearly every disciple from the laws of human perception to describing musical tones. However, in SAS programming a common application of log functions across industries is in the derivation of geometric means. To program a geometric mean, both the log function and its exponential inverse is used.

GEOMETRIC MEANS

Geometric mean can be used to evaluate data covering several orders of magnitude.

The mathematical definition of a geometric mean is the nth root of the product of numbers. However, it's in the practical definition where logarithms are used since the average of the logarithmic values converted back to a base number will give the same result. Example 4 shows this process.

The variables _LOG_ and _LOG10_ are used from Output 1 and a PROC SUMMARY to calculate the mean. Next, the exponential inverse is programmed in a DATA step. In Output 2, GEOMEAN_LOG equals GEOMEAN_LOG10 showing that it does not matter what log base is used. Just make sure that the correct exponent inverse is applied!

Example 4. Geometric Mean Program

```
Proc summary data = TESTDATA NWAY MISSING;
  Var _LOG_ _LOG10_;
  Output out = LOG_MEANS (DROP = _TYPE_ _FREQ_)
         MEAN(_LOG_ _LOG10_) = MEAN_LOG MEAN_LOG10;
Run;

NOTE: There were 10 observations read from the data set WORK.TESTDATA.
NOTE: The data set WORK.LOG_MEANS has 1 observations and 2 variables.
NOTE: PROCEDURE SUMMARY used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds


Data ALL_MEANS;
  Set LOG_MEANS;
  GEOMEAN_LOG = EXP(MEAN_LOG);
  GEOMEAN_LOG10 = 10**MEAN_LOG10;
Run;

NOTE: There were 1 observations read from the data set WORK.LOG_MEANS.
NOTE: The data set WORK.ALL_MEANS has 1 observations and 4 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds
```

Output 2. Geometric Mean Results

	MEAN_LOG	MEAN_LOG10	GEOMEAN_LOG	GEOMEAN_LOG10
1	15.021669088	6.523827994	3340627.0544	3340627.0544

CONCLUSION

There are many applications of logarithms. It is important to make sure that proper function is applied in the analysis. A function's syntax can be misleading so the programmer must be careful that the correct base is used. Having quality checks using the inverse exponential functions or operators can help ensure that the correct log function is programmed.

REFERENCES

1. Logarithms. (August 2014). Retrieved from http://en.wikipedia.org/wiki/Logarithm
2. Lyon, Richard F. (August 2014). Logarithm Plots. Retrieved from http://en.wikipedia.org/wiki/File:Logarithm_plots.png
3. List of Logarithm Identities. (August 2014). Retrieved from http://en.wikipedia.org/wiki/List_of_logarithmic_identities
4. Weisstein, Eric W. (August 2014). Common Logarithm. Retrieved from *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/CommonLogarithm.html
5. Weisstein, Eric W. (August 2014). Common Logarithm. Retrieved from *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/CommonLogarithm.html
6. Costa, Joe. (September 2014). Geometric Mean Calculations. Retreived from http://buzzardsbay.org/geomean.htm

ACKNOWLEDGMENTS

Britney Gilbert:
First, I would like to thank my Lord, Jesus Christ. It is through Him that I find my strength, patience, and resolve. Next, I would like to thank my family: my encouraging husband, Justin, and my kids (Hope, Faith, Justin, Danny, Charity, and Paul) who are my never-ending source of happiness. Finally, I would like to thank Josh Horstman for his great recommendations and collaboration on this paper.

RECOMMENDED READING

Base SAS® Language Reference

CONTACT INFORMATION

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

Britney D. Gilbert
Juniper Tree Consulting, LLC
Britney.Gilbert@JuniperTreeConsulting.com
www.JuniperTreeConsulting.com
@JuniperTree19

Joshua M. Horstman
Nested Loop Consulting
josh@nestedloopconsulting.com
317-815-5899