

Dickens vs. Hemingway: Text Analysis and Readability Statistics in Base SAS®

Jessica Hampton, Cigna Corporation, Bloomfield, CT

ABSTRACT

Although SAS® provides a specific product for text mining (SAS Text Miner), you may be surprised how much text analysis you can readily perform using just Base SAS. The author introduces the topic with some background on widely-used readability statistics and tests in addition to a brief comparison of Hemingway and Dickens. After selecting two appropriate readability tests and texts of similar length, she describes data preparation challenges, including how to deal with punctuation, case, common abbreviations, and sentence segmentation. Using a few simple calculated macro variables, she develops a program which can be re-used to calculate readability tests on any sample input text file. Finally, she validates her SAS output using published readability statistics from sources such as Amazon and searchlit.org.

INTRODUCTION

Charles Dickens and Ernest Hemingway represent two very different writing styles. Dickens (1812 - 1870), a popular Victorian novelist whose writing often included social reform themes, was paid by the word. Hemingway (1899 – 1961), war correspondent-turned-author of American literary classics, was known for his brevity and simplicity of style. These differences can be quantified using common readability statistics and appropriate readability tests.

Readability statistics allow those in various fields (education, healthcare, publishing, military/government, technical documentation) to predict the reading difficulty of a given text or texts. Common readability statistics include the following: total character count, total word count, total sentence count, total paragraph count, average word length, average sentence length, average paragraph length, passive sentence ratio, and distinct word count. Average word length, a measure of semantic difficulty, can be calculated in two different ways: character-based or syllable-based, where the syllable-based approach is harder to derive, and a complex word is typically defined as having three or more syllables. Average sentence length is a measure of syntactic complexity, where longer sentences are typically more complex. Distinct word count gives a measure of semantic variety, but can only be used to compare samples of similar length. Finally, passive sentences are considered more difficult to read than active sentences. Some approaches calculate statistics on the entire text, while others use selected representative or random samples (DuBay, 2004).

Readability tests, based on the readability statistics described above, provide a standardized way to assign a difficulty level or appropriate grade level to a text. The large number of common readability tests includes those listed below: Flesch Reading Ease and Flesch-Kincaid Grade Level Tests, SMOG (Simple Measure of Gobbledygook), Fry Readability Formula, ARI (Automated Readability Index), Coleman-Liau Index, Dale-Chall Readability Formula, Gunning Fog Index, and Kincaid. For more about the history of readability testing, refer to DuBay's *Principles of Readability* (2004).

This paper uses two of the tests listed above, the Automated Readability Index and the Coleman-Liau Index, to compare texts of similar length written by Dickens and Hemingway: *A Christmas Carol* and *Old Man and the Sea*. All programming, including data preparation and sentence segmentation, is developed in Base SAS.

READABILITY TESTS

To give some idea of how readability tests work, we will examine a few in more detail. The Flesch Reading Ease (FRES) and Flesch Kincaid Grade Level (FKGL) Tests are actually available in Microsoft Word® (under Tools>Options>Spelling & Grammar tab> Check grammar with spelling and Show readability statistics). The US Department of Defense uses FRES as a standard test; the calculation is shown below:

$$\text{FRES} = 206.835 - (1.015 \times \text{ASL}^*) - (84.6 \times \text{ASW}^{**})$$

*ASL = average sentence length

**ASW = average word length in syllables

Higher scores indicate easier to read material, whereas lower numbers mark more difficult to read passages. Scores can be interpreted as follows:

90.0 – 100.0: easily understandable by an average 11-year-old student

60.0 – 70.0: easily understandable by 13- to 15-year-old students

0.0 – 30.0: best understood by university graduates

To assign a grade level to a text, calculate FKGL as follows:

$$FKGL = (0.39 \times ASL^*) + (11.8 \times ASW^{**}) - 15.59$$

*ASL = average sentence length

**ASW = average word length in syllables

SMOG is another test which uses a syllable-based measure of semantic difficulty to calculate grade-level difficulty. The SMOG calculator uses a dictionary to look up the syllable length of words, so it counts syllables more accurately than most tests. To approximate the SMOG grade-level formula, perform the following steps: using three 10-sentence examples from the text, count words of three or more syllables, take the square root, and add 3.

The Fry readability formula is an educational tool similarly used to calculate grade level with syllable-based average word length. To calculate a grade level score, first randomly select three separate 100-word passages, counting every word including proper nouns, initializations, and numerals. Next, count the number of sentences in each 100-word sample, estimating to the nearest tenth. Count the number of syllables in each 100-word sample; then plot the average sentence length and the average number of syllables on the Frye graph (not shown). The area in which it falls shows the approximate grade level of the text.

The ARI and Coleman-Liau (CLGL) grade level tests differ from those described above in that they use characters per word instead of syllables per word to estimate semantic difficulty. The ARI formula is shown below:

$$ARI = (4.71 \times AWL^*) + (0.5 \times ASL^{**}) - 21.43$$

*AWL = average word length using number of characters

**ASL = average sentence length

The Coleman-Liau grade level score is calculated as follows:

$$CLGL = (5.89 \times AWL^*) - (30 \times (\# \text{ sentences}/\# \text{ words})) - 15.8$$

*AWL = average word length using number of characters

With so many options, it is important to choose an appropriate test and to use it consistently, since different readability tests can provide a wide range of scores on a single document. Most tests are better suited to certain types of texts than others since these formulas are linear regression equations based on a certain range of data and may not extrapolate well. For example, Flesch is based on school texts from grades 3-12, so it performs best in that range. On the other hand, ARI, based on texts from difficulty levels corresponding to grades 0-7, was originally developed to score US Army technical documents and manuals (Smith & Senter, 1967). Coleman-Liau scores technical documents lower, but is appropriate for 4th grade to college-level texts. For a summary table of various readability tests and their applications, along with selected formulas, see Akerman (2010).

Since *A Christmas Carol* and *Old Man and the Sea* fall within the appropriate difficulty ranges for Coleman-Liau and ARI, in addition to using the more easily calculated character-based average word length, we will select these two formulas for use in SAS programs and test the output against previously published grade-level readability test results for these two texts. Of the two tests, we expect the Coleman-Liau grade level score to be more accurate, given ARI's original application for scoring military technical documents and manuals targeted at an adult audience.

DATA PREPARATION AND SENTENCE SEGMENTATION

Data sets out of copyright, in this case full text files, are easily available from online sources such as Project Gutenberg. There are two aspects to data preparation: word parsing and sentence segmentation (in this case we use a count of words with end punctuation as a proxy rather than actually segmenting and storing a complete list of sentences). To parse words, read in the text file one word at a time, using blank spaces and/or hyphens as delimiters. Punctuation is treated differently when parsing words vs. sentences. For parsing words, remove all punctuation except for apostrophes embedded in words (that is, those which are not leading or trailing). The following program removes double quotes, single quotes, commas, periods, exclamation points, colons, semi-colons, underscores, hyphens, dashes, and asterisks, but retains apostrophes in most possessives and contractions (one exception is contractions at the beginnings of words such as "twas"); exceptions tend to be rare. The reason for retaining apostrophes in contractions is so that words such as "shell" and "she'll" are not grouped together when counting distinct words. However, since periods are used for abbreviations as well as sentence-ending punctuation, some abbreviations may end up being grouped with other words (for example, "U.S." and "us". Depending on the text, additional punctuation may need to be retained or removed, and a more complex solution employing the use of regular expressions may be considered, but for the texts we are examining, this approach is sufficient to obtain a reasonable readability test result. After removing punctuation, make all upper case (or lower case) before grouping to calculate distinct word count. In the example shown below, hyphenated words are treated as separate words, since the program was developed in SAS 9.1. Beginning in SAS 9.2, the DLMSTR option allows use of a multi-character string as a delimiter, but for 9.1 the double hyphen is read as a single hyphen. That is, all hyphens are removed rather than just the double hyphens, or dashes. This means that hyphenated words are treated as multiple words, which may or may not be desirable in each case. For example:

```
data word;
```

```

infile "M:\Jess\&filename..txt" dlm=' ,--';
/*also removes the single hyphen so hyphenated words treated as separate words - dlmstr
starting in 9.2 allows multi-character string as delimiter*/
length word $25.;
input word $ @@;
run;

proc sql;
create table word2 as
select
case when word like "%'" or word like "'%" then compress(word,"'") else word end as
word, /*gets rid of leading or trailing apostrophes*/
length(calculated word) as length,
count(calculated word) as count,
(calculated length) * (calculated count) as chars
from(select
      upcase(compress(word,'!,?.,, ,:,;","(,),_*,')) as word
/* prep for grouping: strip punctuation and upcase*/
      from word
      where calculated word is not null and calculated word <>'')
group by calculated word, calculated length
order by word;

```

For sentence segmentation, it is possible to get a rough estimate of the total number of sentences in the text by counting words ending in periods, exclamation marks, and question marks. Some may note that this method over-estimates the number of sentences due to abbreviations in the text which should not be counted as end of sentence punctuation. This results in shorter average sentence length, which leads to the text being rated at a lower grade level than it should be. A solution to this problem is to deal with abbreviations first before counting end of sentence punctuation. The method shown below uses a common abbreviations match file to eliminate abbreviations, preventing them from being read as end punctuation and inflating the total sentence count.

```

proc sql;
/*can do total words /total sentences to give avg wds/sentence*/
create table sentence as
select
upcase(compress(word,'"'))as word
from word
where word contains '?' or word contains '.' or word contains '!'
;
/*gets rid of abbreviations by matching to file with common abbreviations*/
delete
from sentence
where word in(select * from abbrev)
;
select count(*)into: total_sentences
from sentence
;
quit;

```

CALCULATED MACRO VARIABLES

Using the macro variable &filename to store the name of the text file being analyzed makes the code re-usable. In order to run the analysis on another text sample, replace the filename macro variable with the new file name.

```
%let filename=A_Christmas_Carol;
```

In order to calculate ARI and CLGL, find and store the following macro variables: total word count, total sentence count, total character count. Using the values above, calculate average word length (total characters/total words) and average sentence length (total words/total sentences). Additional calculated variables include word length, distinct word count, ratio of distinct to total word count, and percentage of words greater than 5 characters in length.

```

/*get total distinct words*/
select count(*)into: total_distinct
from word2
;
/*get total words - need for ARI calc*/
select sum(count)into: total_words
from word2
;

```

```

/*get total characters - need for ARI calc*/
select sum(chars)into: total_chars
from word2
;
/*get pct of words at least 5 characters long*/
select pct_total into: pct_long_words
from (select
sum(count) as frequency,
calculated frequency/&total_words as pct_total
from word2
where length >=5)
;

```

Output including ARI and CLGL test results is sent to the SAS log using PUT statements:

```

/*ARI calc and summary statistics*/
data _null_;
avg_wd_len=&total_chars./&total_words;
avg_sentence_len=&total_words./&total_sentences;
ARI=4.71*avg_wd_len + 0.5*avg_sentence_len - 21.43;
CLGL=(5.89*avg_wd_len) - (30*(&total_sentences/&total_words)) - 15.8;
pct_distinct=&total_distinct./&total_words;
%put &filename Readability Statistics;
%put Total Distinct Words = &total_distinct;
%put Total Word Count = &total_words;
put 'Ratio Distinct to Total = ' pct_distinct;
%put Percentage of Words Over 5 Characters Long = &pct_long_words;
%put Total Sentence Count = &total_sentences;
put 'Average Word Length = ' avg_wd_len;
put 'Average Sentence Length = ' avg_sentence_len;
put 'Automated Readability Index (ARI) = ' ARI;
put 'Coleman-Liau Grade Level Index (CLGL) = ' CLGL;
run;

```

Finally, graphical output showing word length vs. word count is produced:

```

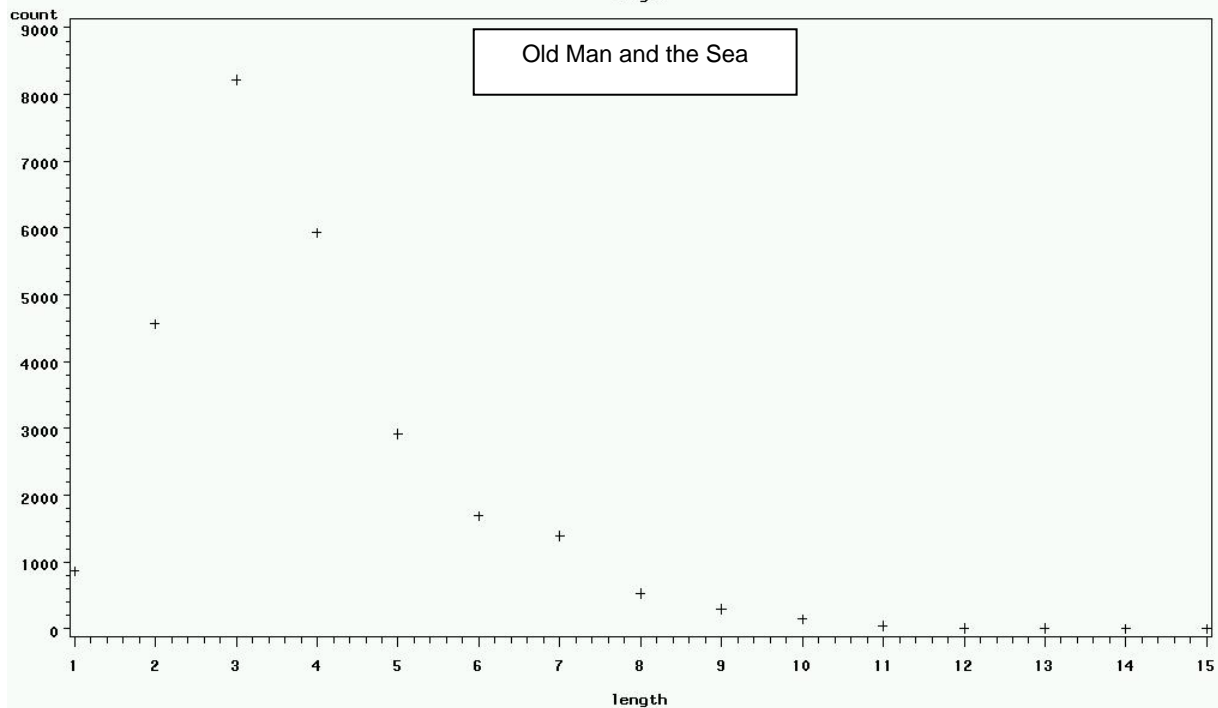
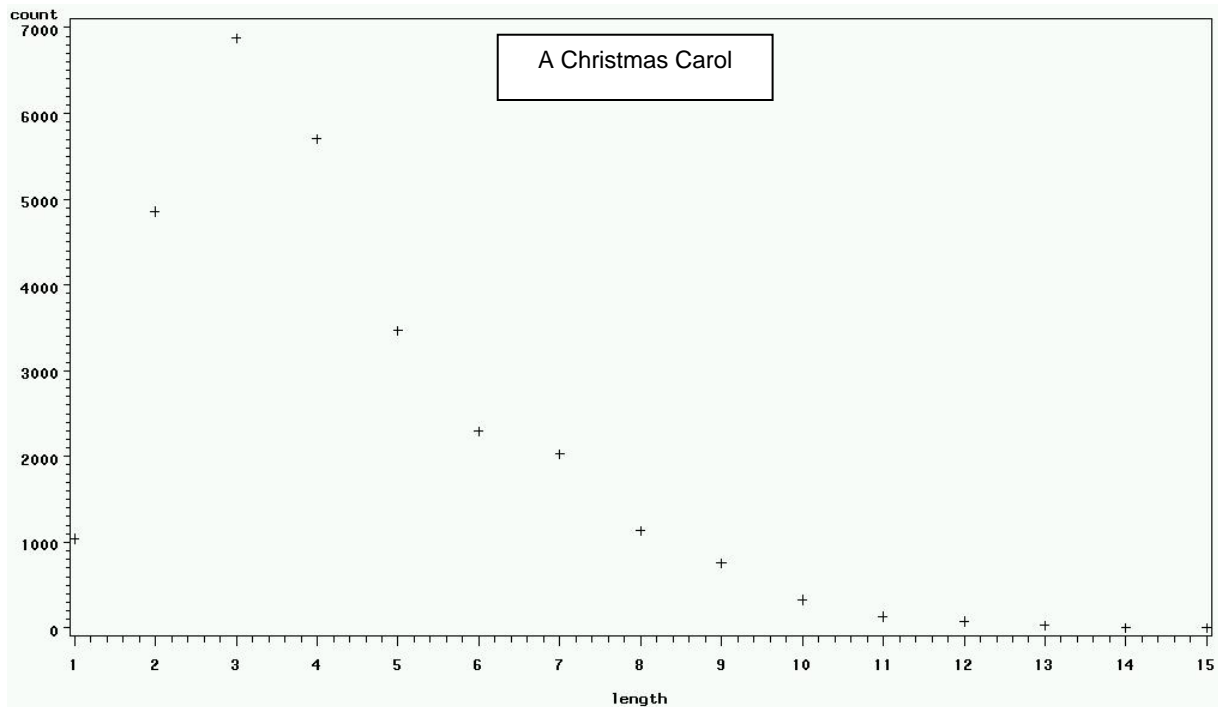
/*group by words of same length and then get as pct of total*/
create table word3 as
select
length,
sum (count) as count,
sum (count)/&total_words format percent6.3 as pct_total
from word2
group by length
order by calculated pct_total desc
;
proc gplot data=word3;
plot count*length;
run;

```

SAS OUTPUT

GRAPHICAL OUTPUT

Gplot output for *A Christmas Carol* and *Old Man and the Sea* is shown on the following page. The plots show that Hemingway's text has a much higher frequency of three-letter words, while Dickens has a higher frequency of words five characters and above. Incidentally, the average word length in English is 4.38 characters, according to a 1923 study by Godfrey Dewey. The study of word length and frequency distributions is a much-studied topic in and of itself; for a history of statistical studies relating word length distributions to the Poisson and other proposed distributions, see Gryzbek (2006). For a discussion of Zipf's law for the relationship between word length and frequency, see Gryzbek (2005). Looking at these plots, we expect that Hemingway has a slightly lower average word length than Dickens. This is consistent with our expectations that Hemingway's text will have a lower difficulty level and less semantic variety than Dickens' text.



LOG OUTPUT

Log output for *A Christmas Carol* is shown below:

```

211 /*ARI calc and summary statistics*/
212 data _null_;
213   avg_wd_len=&total_chars./&total_words;
214   avg_sentence_len=&total_words./&total_sentences;
215   ARI=4.71*avg_wd_len + 0.5*avg_sentence_len - 21.43;
216   CLGL=(5.89*avg_wd_len) - (30*(&total_sentences/&total_words)) - 15.8;
217   pct_distinct=&total_distinct./&total_words;
218   %put &filename Readability Statistics;
A_Christmas_Carol Readability Statistics

```

```

219 %put Total Distinct Words = &total_distinct;
Total Distinct Words =      4319
220 %put Total Word Count = &total_words;
Total Word Count =      28743
221 put 'Ratio Distinct to Total = ' pct_distinct;
222 %put Percentage of Words Over 5 Characters Long = &pct_long_words;
Percentage of Words Over 5 Characters Long = 0.236127
223 %put Total Sentence Count = &total_sentences;
Total Sentence Count =      1891
224 put 'Average Word Length = ' avg_wd_len;
225 put 'Average Sentence Length = ' avg_sentence_len;
226 put 'Automated Readability Index (ARI) = ' ARI;
227 put 'Coleman-Liau Grade Level Index (CLGL) = ' CLGL;
228 run;

```

```

Ratio Distinct to Total = 0.1502626727
Average Word Length = 4.2299690359
Average Sentence Length = 15.199894236
Automated Readability Index (ARI) = 6.0931012772
Coleman-Liau Grade Level Index (CLGL) = 7.1408196778

```

The following table summarizes the output for the analyzed text files:









	Old Man and the Sea	A Christmas Carol	Our Mutual Friend
Total Word Count	26,661	28,743	327,521
Total Distinct Words	2,524	4,319	15,335
Percent Distinct*	9.5%	15.2%	4.7%
Percent Words > 5 Characters	15.6%	23.6%	24.4%
Average Word Length	3.8	4.2	4.2
Average Sentence Length	14.4	15.2	16.5
ARI Grade Level	3.85	6.09	6.79
Coleman-Liau Grade Level	4.72	7.14	7.38

RESULTS









The readability statistics and output shown above confirm expectations for Hemingway vs. Dickens based on author background. Hemingway has less semantic variety, based on total distinct word count and percent distinct words (since *Old Man and the Sea* and *A Christmas Carol* are of similar length, we can use this statistic to compare the two; *Our Mutual Friend* is much longer, and we would have to use a smaller sample if we wanted to compare it with the other two based on distinct word count). Hemingway also has lower average word length than Dickens (proxy for semantic difficulty) and lower average sentence length (proxy for syntactic complexity). The CLGL test places Hemingway at 4th to 5th grade level and Dickens between 6th and 7th grade. The ARI scores are slightly lower for both, but still within a reasonable range. In this situation, we would expect CLGL to be the more accurate test, since ARI was originally designed to score technical manuals.

These results are comparable to published readability test results available online through publisher sources such as Amazon and educator resources such as searchlit.org. Notice the wide range of scores produced on the same text by using different readability tests (results shown below from searchlit.org). Compare the Coleman-Liau results here to those shown in the summary table above (ARI test results are not shown below).

Old Man and the Sea by Hemingway, Ernest - Back to Text Record

Score	Test	Reading Level	Grade (s)	Age (s)	Scale Value	Weight	Words	Source
1	New Dale-Chall	4	4 to 5	9		16.00	<u>6145</u>	oceanstar.com 
2	S/LOG	6	6 to 7	11		12.00	<u>6145</u>	oceanstar.com 
3	Gunning Fog	6.2	6 to 7	11		12.00	<u>6145</u>	oceanstar.com 
4	Coleman-Liau	4.3	4 to 5	9		12.00	<u>6145</u>	oceanstar.com 
5	Laesbarhedsindex (LIX)	4	4 to 5	9	21	12.00	<u>6145</u>	oceanstar.com 
6	Rate Index (RIX)	4	4 to 5	9	1.1	12.00	<u>6145</u>	oceanstar.com 
7	Fry	3	3 to 4	8		12.00	<u>6145</u>	oceanstar.com 
8	Raygor Estimate	3	3 to 4	8		12.00	<u>6145</u>	oceanstar.com 
TOTALS:		4.30	4.75	9.25		100		

Christmas Carol, A by Dickens, Charles - Back to Text Record

Score	Test	Reading Level	Grade (s)	Age (s)	Scale Value	Weight	Words	Source
1	New Dale-Chall	5.5	5 to 6	10		16.00	<u>5928</u>	Marley's Ghost 
2	S/LOG	9.0	9	14		12.00	<u>5928</u>	Marley's Ghost 
3	Gunning Fog	8.8	8	13		12.00	<u>5928</u>	Marley's Ghost 
4	Coleman-Liau	7.5	7	12		12.00	<u>5928</u>	Marley's Ghost 
5	Laesbarhedsindex (LIX)	7	7	12	32	12.00	<u>5928</u>	Marley's Ghost 
6	Rate Index (RIX)	7	7	12	2.7	12.00	<u>5928</u>	Marley's Ghost 
7	Fry	7	7	12		12.00	<u>5928</u>	Marley's Ghost 
8	Raygor Estimate	7	7	12		12.00	<u>5928</u>	Marley's Ghost 
TOTALS:		7.28	7.19	12.13		100		

CONCLUSION

Base SAS may be an option when considering word parsing, sentence segmentation, readability statistics, and other text analysis tasks without having to resort to using or purchasing additional text mining software. Results shown above compare favorably with previously published readability tests from other sources. However, while this approach works well with narrative texts written in standard English where “./?!” have approximate one-to-one correspondence with sentence

boundaries, different customized approaches may be more appropriate for other types of text. For technical manuals written about programming languages where sentence-ending punctuation has additional meanings, a different approach would be needed, possibly one requiring the use of regular expressions. Mathematical texts also use such symbols in formulas and logical expressions. Modern authors often disregard punctuation rules and write in stream-of-consciousness style with ellipses as end punctuation and unclear sentence boundaries. Poetry and narratives written in dialect are additional examples of texts which do not always follow traditional rules. These are just a few examples of why a customized approach is necessary in text analysis. Also it should be noted that what constitutes a sentence, or even a word, can be surprisingly ambiguous and the subject of much debate in linguistic circles. For example, how should nested sentences or hyphenated words be treated? For the example given in this paper, hyphenated words are counted as multiple words since the hyphens are stripped from the text prior to counting words. Similarly, if nested sentences such as dialog end in commas or hyphens, they would not be counted separately, but exclamation points and question marks appearing mid-sentence would be counted. For a discussion of Perl regular expressions and a thoughtful introduction to some of the complexities of text analysis and sentence segmentation, see Chapter 2 of Bilisoly's *Practical Text Mining with Perl* (2008).

REFERENCES:

- Akerman, R. (2010). *Readability Tests and Formulas*. Retrieved from <http://www.ideosity.com/ideosphere/seo-information/readability-tests>, January 11, 2012.
- Bilisoly, R. (2008). *Practical Text Mining with Perl*. Hoboken, New Jersey: John Wiley & Sons.
- Dash, N.S. (2005). *Corpus Linguistics and Language Technology*. New Delhi, India: Mittal Publications.
- Dewey, G. (1923). *Relative Frequency of English Speech Sounds*. Cambridge, Massachusetts: Harvard University Press.
- DuBay, W.H. (2004). *The Principles of Readability*. Costa Mesa, CA. Retrieved from <http://www.impactinformation.com/impactinfo/readability02.pdf>, January 11, 2012.
- Gryzbek, P. (2006). *History and Methodology of Word Length Studies*, Contributions to the Science of Text and Language: Word Length Studies and Related Issues. Pp. 15-90. Retrieved from http://www.peter-grzybek.eu/science/publications/2006/grzybek_2006_history_methodology_word_length.pdf March 2, 1012.
- Gryzbek, P. (2005). *Word Length and Word Frequency*, Contributions to the Science of Text and Language: Word Length Studies and Related Issues. Pp. 277-294. Retrieved from http://www-gewi.uni-graz.at/quanta/publ/quanta_2002/pb_ed/pb_sql-ges.pdf March 2, 1012.
- Smith, E.A., & Senter, R.J. (1967). *Automated Readability Index*. Retrieved from <http://www.dtic.mil/mwg-internal/de5fs23hu73ds/progress?id=KgM/aNDgwc&dl>, January 11, 2012.

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.

Project Gutenberg for free downloads of ebooks: http://www.gutenberg.org/wiki/Main_Page

Readability Statistics on a searchable database of texts: <http://www.searchlit.org/>

CONTACT INFORMATION

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

Jessica Hampton
 CIGNA Corporation
 900 Cottage Grove Rd
 Bloomfield, CT 06002
 Work Phone: (860) 226-1938
 Email: Jessica.Hampton@cigna.com
 Web: <http://www.linkedin.com/in/jessicahampton>
