A DISCUSSION OF METHODS FOR THE DETECTION OF ERRORS IN A POWER LAW DISTRIBUTION

(Research-in-Progress)

Valerie Sessions
Charleston Southern University
vsessions@csuniv.edu

Chris Nuhn
Charleston Southern University
ccclxspyder@hotmail.com

Abstract: To detect erroneous data points within a set, one must first understand how the data is usually distributed. Often in our analysis of data, we assume a data set to be normal (that is to follow a normal distribution or be normally distributed). This often is not the case, however, particularly when dealing with human-centric events such as web site hits, word usage, or author citations. In this paper, we examine the power law distribution and the types of data that should be tested for such a distribution. We present test methods and results from the testing of the robustness of a power law distribution under conditions of randomly generated errors in the data set, and discuss methods for error detection in data sets with this type of distribution. We conclude with a discussion of future research.

Key Words: Data Quality, Information Quality, IQ Concepts, Metrics, Measures, and Models

INTRODUCTION

One often begins an analysis of a set of data by plotting the variables as a graph, and determining the mean, median and variance of the data. If the distribution is found to be normal, we unlock a variety of useful functions for dealing with this data. Normal distributions are the most widely used model for random variables with a continuous distribution and often it is a "mathematical convenience to assume that the distribution from which a random sample is drawn is a normal distribution [2]". While other distributions are also popular and widespread in physical science and human events, often we first consider a normal distribution as a fit for the data set. As we will see in our subsequent analysis, this can obscure the true distribution of the data and therefore meaningful further analysis of outliers or variances in the data set.

As an illustration, we take the data in Table 1 below.

Number	Count	Number	Count
1	214	28	1145
2	227	29	946
3	233	30	840
4	261	31	645
5	277	32	553
6	288	33	544
7	297	34	488
8	303	35	486
9	309	36	414
10	352	37	403
11	366	38	381
12	368	39	375
13	374	40	353
14	403	41	349
15	416	42	340
16	516	43	300
17	525	44	296
18	531	45	286
19	584	46	262
20	663	47	251
21	846	48	239
22	934	49	224
23	1101	50	223
24	2268	51	218
25	2006	52	217
26	8024	53	203
27	5184	54	201

Table 1: Data Counts

Upon initial graphing of this data we see a distribution that appears normal, see Figure 1.

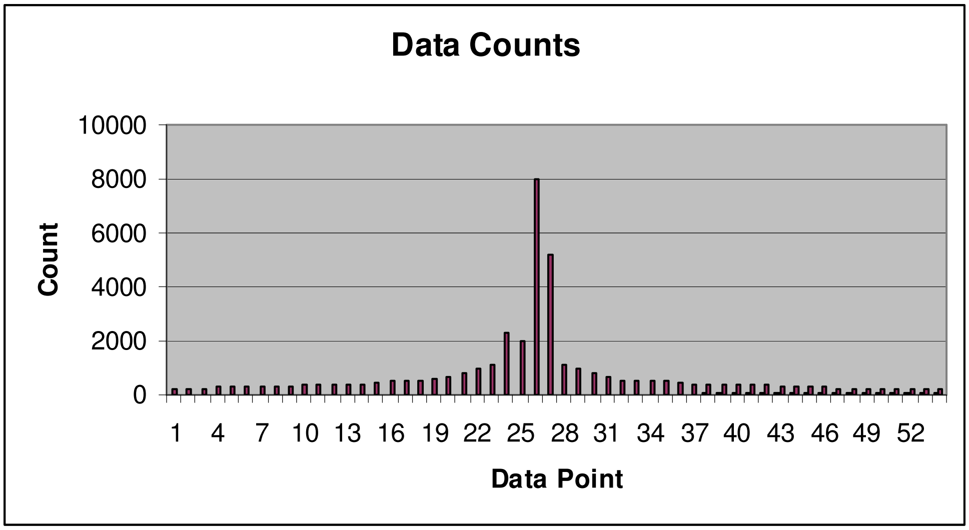

Figure 1: Initial Graphing of data count

Upon examination of mean, median, and mode however we find that the three are close but are not equal.

mean	median	mode
723	371	403

Undaunted we consider the high center point – (#26 – 8024) a probable outlier and the numbers become closer to a normal distribution, as does the graph.

mean	median	mode
497	367	403

Because it is simply more convenient, we then consider the distribution normal, determine the standard deviation and consider all points outside of three standard deviations outliers. In either case above (dropping or keeping data point #26 in our calculation), we can consider the highest value as an outlier [1].

All of this is nonsense and an example of poor statistical analysis (however common it may be), because as we will see in the discussion below, this set of data is actually linguistic and follows a Zipfian distribution. But for the sake of discussion and illustration, let us look at what would happen if we misclassify the data as normal and have an error occur in the data set. If we erroneously increase data point #21 by a factor of 2 it becomes 1692 and in either scenario this falls within an acceptable range for this distribution and would not be flagged as an outlier.

As we have already discussed, however, this data actually represents word counts inputted to ALICE [11], in the authors' opinion a fantastic Chat Bot developed by Dr. Richard S. Wallace of the ALICE Artificial Intelligence foundation [2]. By understanding that the data is linguistic in nature, we would naturally experiment with a Zipfian or power law distribution instead. Ranking and then plotting the data on a log-log graph we discover the power law distribution of the data as shown in Table 2 and Figure 2.

[1] In the first case $\sigma = 1261$, $3\ \sigma = 3782$ and in the second case $\sigma = 405$, $3\ \sigma = 1711$.

[2] http://alicebot.blogspot.com/

Rank	Count	Rank	Count
1	8024	28	368
2	5184	29	366
3	2268	30	353
4	2006	31	352
5	1145	32	349
6	1101	33	340
7	946	34	309
8	934	35	303
9	846	36	300
10	840	37	297
11	663	38	296
12	645	39	288
13	584	40	286
14	553	41	277
15	544	42	262
16	531	43	261
17	525	44	251
18	516	45	239
19	488	46	233
20	486	47	227
21	416	48	224
22	414	49	223
23	403	50	218
24	403	51	217
25	381	52	214
26	375	53	203
27	374	54	201

Table 2: Ranked Data and Count

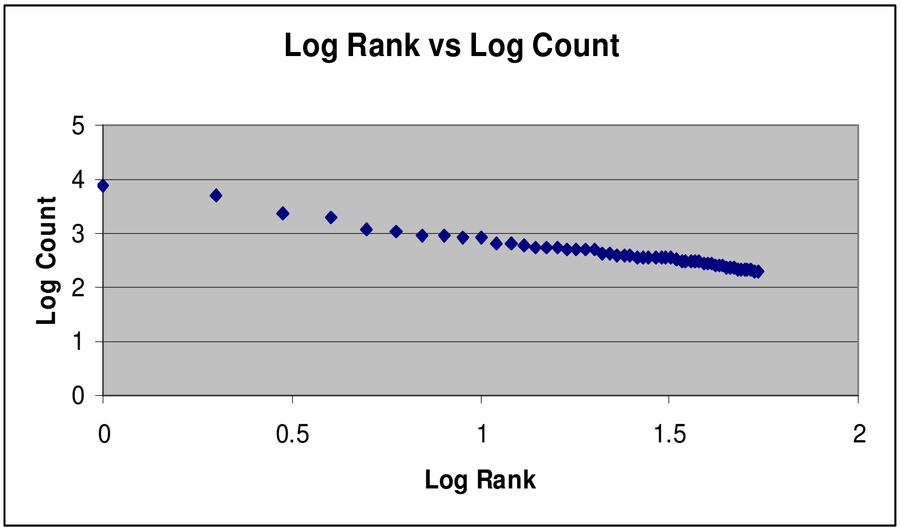

Figure 2: Distribution of data

In this plotting and understanding of the data, if data point #21 (count 846) were to suddenly double, this would easily be highlighted in the graph and brought to the attention of the data analyst.

Note: This was an illustrative example of the errors that may occur in assuming a data set is normally distributed and does not represent how the ALICE AI Foundation used or analyzed the data above.

While determination of outliers within a normal distribution is commonplace and well represented in the literature, determination of outliers (perhaps errors in the data) in power law distributions is less popular in the literature. In our research, we will attempt to utilize the power law distribution to uncover data inaccuracies in experimental data sets that follow a power law distribution. Our goal is to uncover inaccuracies that might not be easily recognized when data is plotted as a normal distribution. As we present here, during this phase of our research we test the robustness of a power law distribution under randomly generated errors in the set and explore methods for the determination of inaccurate sets.

REVIEW OF LITERATURE

A normal distribution is a special type of standard distribution which follows a bell-shaped curve pattern over a set of data. This bell-shaped curve has several key properties. The distribution is always symmetric, it is concentrated in the center and decreases rapidly on either side, meaning that the data has less frequent near extreme values. The area under the curve has a direct correlation to the probability that that value has of occurring. A large area implies a large probability and a small area implies a small probability [9].
Normal distributions follow the form:

$$F(x) = \frac{e^{\frac{-(x-\mu)^2}{(2\sigma^2)}}}{\sigma\sqrt{2\pi}}$$

, where x is the variant with mean μ and variance σ^2.
Data that follow a normal distribution obey the central limit theorem, which states that for any distribution with mean μ and variance σ^2, the distribution approaches normal as the sample size continues to increase.
After determining that a function is normal, outliers can then be calculated using Grubb's Test [4]. According to this test, any data point located beyond $(Y - Y_{\min})/s$ or $(Y - Y_{\max})/s$, where Y is the sample mean and s is the standard deviation, is considered an outlier.
While the normal distribution is often assumed, other distributions do exist, including the power law distribution. A power law distribution is a mathematical relationship between two quantities where the number or frequency of an event varies as a power of some attribute of that object. The law normally appears in the form:
$F(x) = ax^k + o(x^k)$, where a and k are constants, $o(x^k)$ is an asymptotically small function of x^k, and k is a scaling exponent [7].
Pareto is the most famous for using this law with his "80-20" theory, which states that 20% of the population holds 80% of the wealth. The graph of this power rank function looks close to exponential, although reversed. Showing that a small percentage contains most of the wealth and as the percentage increases, the distribution falls drastically until it bottoms out along the x-axis. [5].
Zipf's Law is another special type of power law distribution, where the constants are zero. Harvard linguist George Kingsley Zipf observed that if values are given ranks in descending order, that the values are inversely proportional to their ranks compared to the highest rank. That is, the rank two value occurs ½ as much as rank one, while rank three occurs 1/3 as often.
According to Zipf's law, when the log of the rank is plotted against the log of the values, a line is formed. This is very helpful when determining if a distribution does indeed follow Zipf's law. To determine this,

the correlation coefficient is calculated using the logarithmic values of the ranks and data. This coefficient is a number ranging from [-1,1] where the closer to one, or negative one, the value is, the closer it is to a line. While originally used with linguistic data sets, Zipf's law has been influential in many fields, including computer music [3].

Determining that a data set displays a power law distribution is a challenging research question. The authors shall summarize here pivotal work in this field by Caluset et al [1]. In their research, Clauset et al summarize several data sets that are assumed to have power law distributions, among these count of word use from Moby Dick, book sales, papers authored, and population of cities. They then summarize the most popular method of categorizing a data set as having a power law distribution, which we explained above – plotting the log of the count by the log of the rank of the values and determining through linear regression if the values create a line. There are problems with using this naive method for determining a power law distribution as explained in Clauset et al. Their research goes further and the team creates its own methods for determining a power rank distribution. For a thorough summary of the power law and Clauset et al's work we encourage the reader to view the paper in its entirety. While we did not use their particular methods and algorithms instead maintaining the usual method of plotting a log-log graph and determining the slope and correlation coefficient, we did rely on their research to determine which data sets should be considered to have follow a power law distribution.

Methods for the Detection of Inaccurate Data Sets

Assuming a power law distribution has been determined, we wish to have methods for the detection of errors within the set (or errors in new data points coming into the set). We present and test three methods for determining inaccuracies in this type of distribution:

1. Value of the slope of the line created by plotting the *log(rank) log(count*) of the data variables.

2. Change in the value of the slope of the line created by plotting the *log(rank) log(count*) of the data variables.

3. Change in the correlation coefficient values (how closely the data fit a line).

In this initial research we chose to look at the slope of the resulting line, change in the slope from a gold standard data set, and to analyze any changes in the correlation coefficient (how closely the data fit a line). In order to test these methods and determine how data sets with a power law distribution handle errors in the data, we create the following test data sets and perform our tests.

Creation of Test Data Sets

In order to test the robustness of data sets conforming to a power law distribution under situations of inaccurate data, we created test sets with randomly corrupted data in differing percentages. For instance, we take our initial data set, which follows a power law distribution, and label it the Gold Standard data set. We then use our random number generator to produce erroneous data in differing percentages, 1%, 3%, 15%. etc. We then test the data to determine the resulting slope and correlation coefficients.

In order to create random erroneous data, we corrupted the data using two methods. In the first method, we used the Perl random number generator with a range from 0 to 10 percent greater than the highest value in the particular data set. This appeared to skew the random numbers higher than would be likely in erroneous data, so we also created a second method where we took the randomly generated value between 0 and 10 percent above the highest value, and used that as the top value to randomly generate a second number.

Results from both types of randomly generated data errors followed a similar pattern. See our Perl script in Appendix B.

Our Gold Standard data sets were chosen based on their conformity to a power law distribution as outlined in [1]. We chose sets with p-values greater than 0.10 corresponding to a statistically significant power-law fit. For our research we chose the following data sets and would like to thank Clauset et al for providing links and resources for these and other possible power law distribution data sets on their website:

A) Word Count – this is a count of the frequency of unique words from the novel Moby Dick by Herman Melville [9].

B) Solar Flares – count of the peak intensity of solar flares between 1980-1989 [9].

C) Names – the frequency of the occurrence of family surnames in the US 1990 census [10].

D) Citations – Author citations in June 1997 by authors published in 1981 [6].

E) Religious Affiliation – religious affiliation in the US as reported by the website adherents.com [8].

Each of the above data sets was considered the Gold Standard set, and was corrupted in different percentages by the two random number generation methods as described above.

TEST RESULTS

Our test results are presented below in chart and graphical form. The results for change in slope were not promising and are therefore shown only in Appendix A. Also, due to space limitations, in Appendix A we show full results for only one of the two random error generation methods because the results were similar. The full results can be obtained by emailing the authors.

Percent Corrupt	R - squared value Random Method I				
	Word Count	Solar Flares	Family Name	Citations	Religious
Gold Standard - 0%	-0.988579	-0.992570	-0.994842	-0.987608	-0.939689
1%	-0.977787	-0.977897	-0.997266	-0.992400	-0.939689
2%	-0.958271	-0.960423	-0.993554	-0.990630	-0.939689
3%	-0.946614	-0.948668	-0.987005	-0.981541	-0.939689
4%	-0.937757	-0.939553	-0.975075	-0.966808	-0.939689
5%	-0.926703	-0.928075	-0.962358	-0.945712	-0.916498
6%	-0.911636	-0.913613	-0.945028	-0.925201	-0.912175
7%	-0.892204	-0.895342	-0.926630	-0.902501	-0.905218
8%	-0.870329	-0.872613	-0.903462	-0.873514	-0.884737
9%	-0.857065	-0.847017	-0.879958	-0.839687	-0.884299
10%	-0.837566	-0.819033	-0.853686	-0.806561	-0.872218
11%	-0.814163	-0.790224	-0.832537	-0.766355	-0.809960
12%	-0.788848	-0.760147	-0.810317	-0.726180	-0.761718
13%	-0.762256	-0.729522	-0.785785	-0.692132	-0.751060
14%	-0.733374	-0.700011	-0.759144	-0.660768	-0.740559
15%	-0.706243	-0.670777	-0.738759	-0.621011	-0.745369
20%	-0.673616	-0.641851	-0.714443	-0.587464	-0.774571
25%	-0.638245	-0.615828	-0.689070	-0.548562	-0.712225
30%	-0.606023	-0.587774	-0.663526	-0.509061	-0.714426
35%	-0.612337	-0.567309	-0.638208	-0.480668	-0.767803
40%	-0.614250	-0.552448	-0.624336	-0.467117	-0.753158
45%	-0.636806	-0.551619	-0.621493	-0.466787	-0.776767
50%	-0.667539	-0.559030	-0.634528	-0.502952	-0.712290

Table 3: Overall results Correlation Coefficient Change (r-squared value) for errors generated with random generation method I

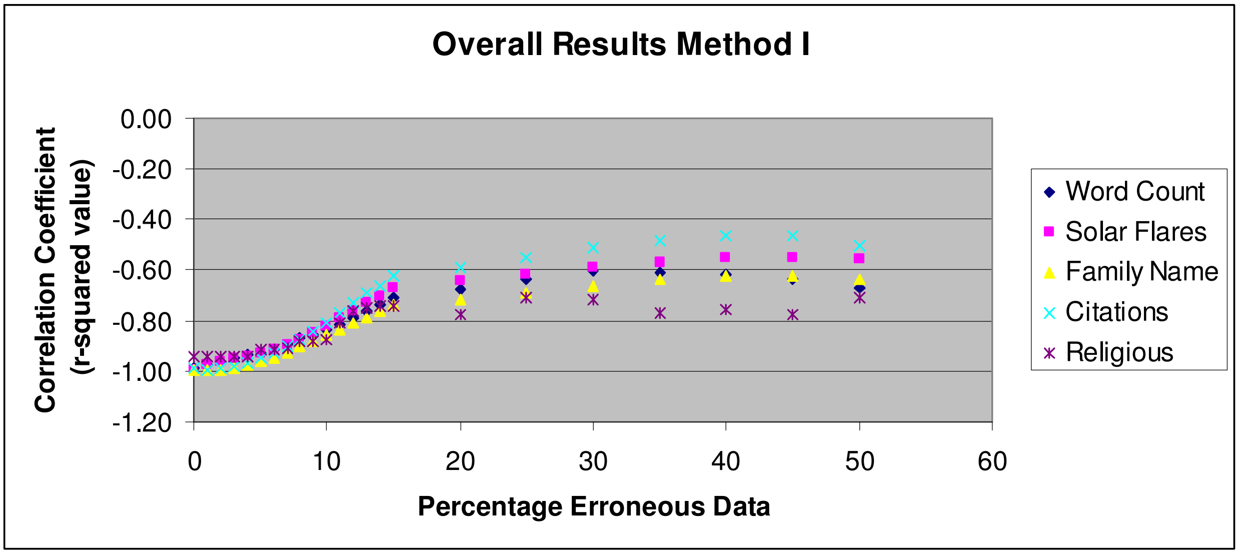

Figure 3: Overall Results Method I

Percent Corrupt	**R - squared value Random Method II**				
	Word Count	**Solar Flares**	**Family Name**	**Citations**	**Religious**
Gold Standard - 0%	-0.988579	-0.992570	-0.994842	-0.987608	-0.939689
1%	-0.978572	-0.981384	-0.995643	-0.992367	-0.939689
2%	-0.961307	-0.966405	-0.993760	-0.990391	-0.939689
3%	-0.950432	-0.954958	-0.986636	-0.981653	-0.939689
4%	-0.941061	-0.944805	-0.975702	-0.966876	-0.939689
5%	-0.929961	-0.933324	-0.961673	-0.947337	-0.925849
6%	-0.915121	-0.918234	-0.945889	-0.922518	-0.905163
7%	-0.896919	-0.899130	-0.927446	-0.899288	-0.893834
8%	-0.873743	-0.876048	-0.905224	-0.869428	-0.869612
9%	-0.847091	-0.852133	-0.881373	-0.834736	-0.856037
10%	-0.818618	-0.824686	-0.856301	-0.806868	-0.855829
11%	-0.787823	-0.794837	-0.834615	-0.771993	-0.827903
12%	-0.756305	-0.765638	-0.811573	-0.738513	-0.813643
13%	-0.726605	-0.735462	-0.784663	-0.695729	-0.765959
14%	-0.695377	-0.706243	-0.760018	-0.659897	-0.756154
15%	-0.665054	-0.677629	-0.737292	-0.631092	-0.741744
20%	-0.630705	-0.643865	-0.708544	-0.597312	-0.759030
25%	-0.597631	-0.612905	-0.686275	-0.548669	-0.702058
30%	-0.565641	-0.584952	-0.666811	-0.493476	-0.750499
35%	-0.536420	-0.560730	-0.652261	-0.458648	-0.702436
40%	-0.517770	-0.547288	-0.642082	-0.446062	-0.695310
45%	-0.509191	-0.547264	-0.637989	-0.452451	-0.582652
50%	-0.512050	-0.558067	-0.644999	-0.469800	-0.403287

Table 4: Overall results Correlation Coefficient Change (r-squared value) for errors generated with random generation method II

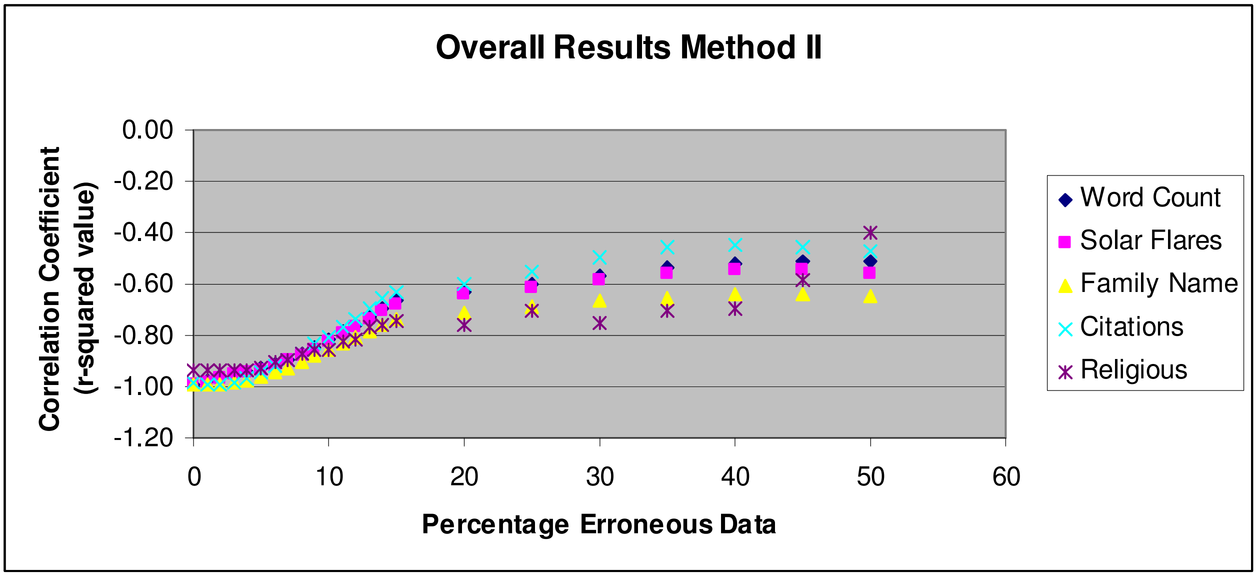

Figure 4: Overall Results Method II

DISCUSSION OF RESULTS

There are many aspects of our test results to comment on. First, it does not appear to significantly change the resulting distribution of slope values or correlation coefficients regardless of which of the two random error generating methods were used. Second, it appears that Methods I and II, slope and change in slope respectively, do not yield quantifiable results. For example, looking at the results in Appendix A, Table 5, one notices that while the slope changes at a predictable rate based on percentage of erroneous data, at 45% inaccurate data the slope itself is very close to the Gold Standard slope. Difference between slope and gold standard slope also do not contain predictable measurements that are useful for our purposes.

The third method, change in correlation coefficient, or r-squared value, however does yield predictable and useful measurements for determining the level of inaccuracies in the data set. We can see from Tables 3 and 4, and Figures 3 and 4, that the correlation coefficient gets farther away from a line (farther from -1 or 1) as the amount of inaccurate data increases. Furthermore, in all but data set E the correlation coefficient values are even similar among the various data sets. These are promising results for many reasons. The slope of the line does change as the amount of inaccurate data increases, however one need not know the original slope of the line to determine the amount of inaccuracy. Instead, if the data is hypothesized to be a power law distribution, it can be plotted and fitted to a line regardless of the slope and an estimate of the level of inaccuracies within the data set can be made.

The gradual nature of the change in the correlation coefficient also suggests that the change could be fitted to a defined correlation between level of inaccuracy and coefficient value so that estimations of the quality of the set could be made. It is too early in our research to strictly define this, but it is interesting that this type of data appears to follow a steady decline as opposed to other sets that have a severe deterioration of distribution with even small amounts of inaccuracies as we have seen in other work [8]. In most instances the data sets remain within a 0.8 r-squared value until around 12% inaccurate data. This is very encouraging for the creation of a robust inaccuracy estimator.

CONCLUSIONS AND FUTURE WORK

This research is still in its early phases. Ultimately the authors would like to test approximately twenty data sets and determine if a definite correlation between r–squared value and level of inaccuracy does indeed exist, and if so to define this. There is also a fourth method that may be promising but that we have not tested that would allow us to determine if there as been an error in one particular data count. If only one value has a perceived error we could account for this by determining if its rank or count changes dramatically over the course of some time period. If there is a dramatic increase this could be flagged and analyzed further. Finally, use of the new methods of Clauset et al [1] and the determination of methods for determining errors using their algorithms could be investigated. Overall, the authors are encouraged by these initial results and will pursue this avenue of research further as many data sets may conform to a power law distribution.

REFERENCES

[1] Clauset, A, C.R. Shalizi, and M.E.J. Newman. (2009) Power-law distributions in empirical data. *SIAM Review* 51, 661-703.

[2] DeGroot, M. H. and Mark Schervish. (2002). Probability and Statistics. 3rd Edition. Addison Wesley. Boston.

[3] Manaris, B. J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. Davis. (2005). "Zipf's Law, Music Classification, and Aesthetics".*Computer Music Journal.* **29** (1). 55-69.

[4] National Institute of Standards and Technology. Engineering Statistics Handbook. Section 1.5.3.17.

[5] Newman, M.E.J. (2005). "Power laws, Pareto distributions and Zipf's law." *Contemporary Physics* **46**, 323.

[6] Redner, S. (1998). "How Popular is Your Paper? An Empirical Study of the Citation Distribution." *European Physical Journal B* **4**, 131 .

[7] Reed, W. (2001) The Pareto, Zipf and other power laws. Economics Letters, Volume 74, Issue 1.

[8] Sessions, V. and M. Valtorta. (2009). "Towards a Method for Data Accuracy Assessment Utilizing a Bayesian Network Learning Algorithm." Journal of Data and Information Quality (JDIQ). 1 (3).

[9] Simon, S. (2008) What is a normal distribution? http://www.cmh.edu/stats/definitions/norm_dist.htm

[10] US Census Bureau, 1990 US Census, Summary Tape File 1 (STF1) and Summary Tape File 3 (STF3).

[11] Wallace, R. "Zipf's Law", http://www.alicebot.org/articles/wallace/zipf.html

Appendix A Individual Results

Percent Corrupt	Slope	r squared value	Distance from Gold Standard Slope
Gold Standard (0%)	-1.113275	-0.988579	0
1%	-1.330470	-0.977787	0.217195
2%	-1.658394	-0.958271	0.545119
3%	-2.025777	-0.946614	0.912502
4%	-2.383607	-0.937757	1.270332
5%	-2.706081	-0.926703	1.592806
6%	-2.969563	-0.911636	1.856288
7%	-3.157777	-0.892204	2.044502
8%	-3.266302	-0.870329	2.153027
9%	-3.252679	-0.857065	2.139403
10%	-3.192771	-0.837566	2.079495
11%	-3.085507	-0.814163	1.972231
12%	-2.947045	-0.788848	1.833770
13%	-2.783619	-0.762256	1.670344
14%	-2.598766	-0.733374	1.485491
15%	-2.410874	-0.706243	1.297599
20%	-2.173447	-0.673616	1.060172
25%	-1.912633	-0.638245	0.799358
30%	-1.661293	-0.606023	0.548018
35%	-1.484651	-0.612337	0.371376
40%	-1.298752	-0.614250	0.185476
45%	-1.184591	-0.636806	0.071315
50%	-1.091816	-0.667539	-0.021460

Table 5: Data Set A Tabular Results

Percent Corrupt	Slope	r squared value	Distance from Gold Standard Slope
Gold Standard (0%)	-1.069828	-0.992570	0.000000
1%	-1.239464	-0.977897	-0.169636
2%	-1.497125	-0.960423	-0.427297
3%	-1.796500	-0.948668	-0.726672
4%	-2.099655	-0.939553	-1.029827
5%	-2.367906	-0.928075	-1.298078
6%	-2.580713	-0.913613	-1.510885
7%	-2.732337	-0.895342	-1.662509
8%	-2.823911	-0.872613	-1.754083
9%	-2.853433	-0.847017	-1.783605
10%	-2.825839	-0.819033	-1.756011
11%	-2.751598	-0.790224	-1.681770
12%	-2.636783	-0.760147	-1.566955
13%	-2.497380	-0.729522	-1.427552
14%	-2.340690	-0.700011	-1.270862
15%	-2.172721	-0.670777	-1.102893
20%	-1.955555	-0.641851	-0.885727
25%	-1.735631	-0.615828	-0.665803
30%	-1.514631	-0.587774	-0.444803
35%	-1.299938	-0.567309	-0.230110
40%	-1.106034	-0.552448	-0.036206
45%	-0.964522	-0.551619	0.105306
50%	-0.863613	-0.559030	0.206215

Table 6: Data Set B, Tabular Results

Percent Corrupt	Slope	r squared value	Distance from Gold Standard Slope
Gold Standard (0%)	-0.852299	-0.994842	0.000000
1%	-0.920012	-0.997266	-0.067713
2%	-1.042925	-0.993554	-0.190626
3%	-1.168022	-0.987005	-0.315723
4%	-1.315955	-0.975075	-0.463656
5%	-1.447085	-0.962358	-0.594786
6%	-1.564527	-0.945028	-0.712228
7%	-1.640417	-0.926630	-0.788118
8%	-1.690200	-0.903462	-0.837901
9%	-1.700633	-0.879958	-0.848334
10%	-1.691585	-0.853686	-0.839286
11%	-1.655821	-0.832537	-0.803522
12%	-1.595099	-0.810317	-0.742800
13%	-1.524150	-0.785785	-0.671851
14%	-1.446234	-0.759144	-0.593935
15%	-1.368551	-0.738759	-0.516252
20%	-1.267139	-0.714443	-0.414840
25%	-1.151323	-0.689070	-0.299024
30%	-1.045189	-0.663526	-0.192890
35%	-0.927194	-0.638208	-0.074895
40%	-0.819479	-0.624336	0.032820
45%	-0.755504	-0.621493	0.096795
50%	-0.712343	-0.634528	0.139956

Table 7: Data Set C, Tabular Results

Percent Corrupt	Slope	r squared value	Distance from Gold Standard Slope
Gold Standard (0%)	-2.497718	-0.987608	0.000000
1%	-2.678983	-0.992400	-0.181265
2%	-2.936413	-0.990630	-0.438695
3%	-3.289699	-0.981541	-0.791981
4%	-3.640914	-0.966808	-1.143196
5%	-3.961571	-0.945712	-1.463853
6%	-4.167570	-0.925201	-1.669852
7%	-4.306592	-0.902501	-1.808874
8%	-4.379137	-0.873514	-1.881419
9%	-4.370368	-0.839687	-1.872650
10%	-4.260689	-0.806561	-1.762971
11%	-4.073511	-0.766355	-1.575793
12%	-3.841621	-0.726180	-1.343903
13%	-3.599693	-0.692132	-1.101975
14%	-3.311202	-0.660768	-0.813484
15%	-2.988006	-0.621011	-0.490288
20%	-2.681185	-0.587464	-0.183467
25%	-2.341687	-0.548562	0.156031
30%	-1.927895	-0.509061	0.569823
35%	-1.602143	-0.480668	0.895575
40%	-1.346614	-0.467117	1.151104
45%	-1.111651	-0.466787	1.386067
50%	-1.023600	-0.502952	1.474118

Table 8: Data Set D, Tabular Results

Percent Corrupt	Slope	r squared value	Distance from Gold Standard Slope
Gold Standard (0%)	-3.170003905	-0.939689	0.000000
1%	-3.170003905	-0.939689	0.000000
2%	-3.170003905	-0.939689	0.000000
3%	-3.170003905	-0.939689	0.000000
4%	-3.170003905	-0.939689	0.000000
5%	-3.196891387	-0.916498	-0.026887
6%	-3.166925523	-0.912175	0.003078
7%	-3.187790214	-0.905218	-0.017786
8%	-3.028894425	-0.884737	0.141109
9%	-3.025066111	-0.884299	0.144938
10%	-3.036999758	-0.872218	0.133004
11%	-2.80998711	-0.809960	0.360017
12%	-2.677691993	-0.761718	0.492312
13%	-2.667245636	-0.751060	0.502758
14%	-2.653013085	-0.740559	0.516991
15%	-2.634523768	-0.745369	0.535480
20%	-2.668712705	-0.774571	0.501291
25%	-2.089574955	-0.712225	1.080429
30%	-2.046534647	-0.714426	1.123469
35%	-1.84755318	-0.767803	1.322451
40%	-1.263565195	-0.753158	1.906439
45%	-1.015029791	-0.776767	2.154974
50%	-1.121950978	-0.712290	2.048053

Table 9: Data Set E, Tabular Results

Appendix B

Sample Perl scripts. For more source code contact the authors. This was compiled under Perl 5.10.1.

```
#Chris Nuhn & Doc Sessions
#6:00 PM 02/16/2010
#Zipf's Powerlaw v3.2

####################################
#            ARGS                  #
####################################
# 1. String         File           #
# 2. Boolean        Create .xls    #
# 3. Boolean        Print          #
####################################

if ($#ARGV >= 2) {$print = $ARGV[2];}
if ($#ARGV >= 1) {$xls = $ARGV[1];}
if ($#ARGV >= 0) {$file = $ARGV[0]; $error = "$file not found\n";}
if ($#ARGV <  0) {$error = "file not given\n";}

open (IN, $file) or die $error;
$rank = 0;
foreach (<IN>)
{
	chomp $_;
	next unless $_ =~ m/\t*(\d+)\t*/;
	$y[$rank] = $1;
	$rank++;
}
```

```
@y = sort {$b <=> $a} @y;

$rank = 0;
foreach (@y)
{
        $logy[$rank] = log10($_);
        $rank++;
}

foreach (1..($#y+1))
{
        $x[$_ - 1] = $_;
        $logx[$_ - 1] = log10($_);
}

correlation(\@logx, \@logy, $print);
if($xls) {toxls();}

###########################################
#            SUBRUTINES                    #
###########################################
sub mean
{
        my ($ref_) = @_;
        my $size = @{$ref_};
        my $mean = 0;
        foreach (@{$ref_})
        {
             $mean += $_;
        }
        return ($mean / $size);
}

sub toxls()
{
        ($first, $extension) = split /\./, $file;
        open( OUT, ">$first"."log.xls");
        foreach (0..$#y)
        {
             print OUT $x[$_], "\t", $y[$_], "\t", $logx[$_], "\t", $logy[$_], "\n";
        }
        print "$first"."log.xls created\n";
}

sub sd
{
        my ($ref_) = @_;
        my $size = @{$ref_};
        my $mean = mean($ref_);
        my $sqtotal = 0;
        foreach (@{$ref_})
        {
```

```
        $sqtotal += sqr($_ - $mean);
    }
    return sqrt($sqtotal / $size);
}

sub correlation
{
    my ($ref_x, $ref_y, $print) = @_;
    my $n = @{$ref_x};
    my $sdx = sd($ref_x);
    my $sdy = sd($ref_y);
    my $xbar = mean($ref_x);
    my $ybar = mean($ref_x);
    my $sum = 0;
    foreach (0..($n - 1))
    {
        $sum += ((@{$ref_x}[$_] - $xbar)*(@{$ref_y}[$_] - $ybar));
    }
    my $corr = ($sum / (($n) * $sdx * $sdy));
    if($print) {print "Correlation:\nr: $corr\n";}
    return $corr;
}

sub sqr
{
    return (@_[0] * @_[0]);
}

sub log10
{
    my ($num) = @_;
    return (log($num) / log(10))
}
```