ITECH1400 Foundations of Programming

Logarithms, Benford's Law and Fraudulent Data

Overview

In this assignment you will write an application in Python that will apply Benford's Law to a given set of your own data. This is an individual assignment.

Timelines and Expectations

Percentage Value of Task: 20%

Due: Friday 5 June 2020 @17:00 (week 11)

Minimum time expectation: 20 hours

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

K1. Identify and use the correct syntax of a common programming language.

K2. Recall and use typical programming constructs to design and implement simple software solutions.

K3. Reproduce and adapt commonly used basic algorithms.

K4. Explain the importance of programming style concepts (documentation, mnemonic names, indentation)

S2. Write and implement a solution algorithm using basic programming constructs.

S3. Demonstrate debugging and testing skills whilst writing code.

A1. Develop self-reliance and judgement in adapting algorithms to diverse contexts.

A2. Design and write program solutions to identified problems using accepted design constructs.

Assessment Details

Background

At the turn of the century, if you wanted to do some serious calculating you usually used a book of logarithms[1] to help you do the arithmetic; the pages in the book of logarithms were arranged in numerical order.

A sample logarithms page, from the link below, is shown here on the right.

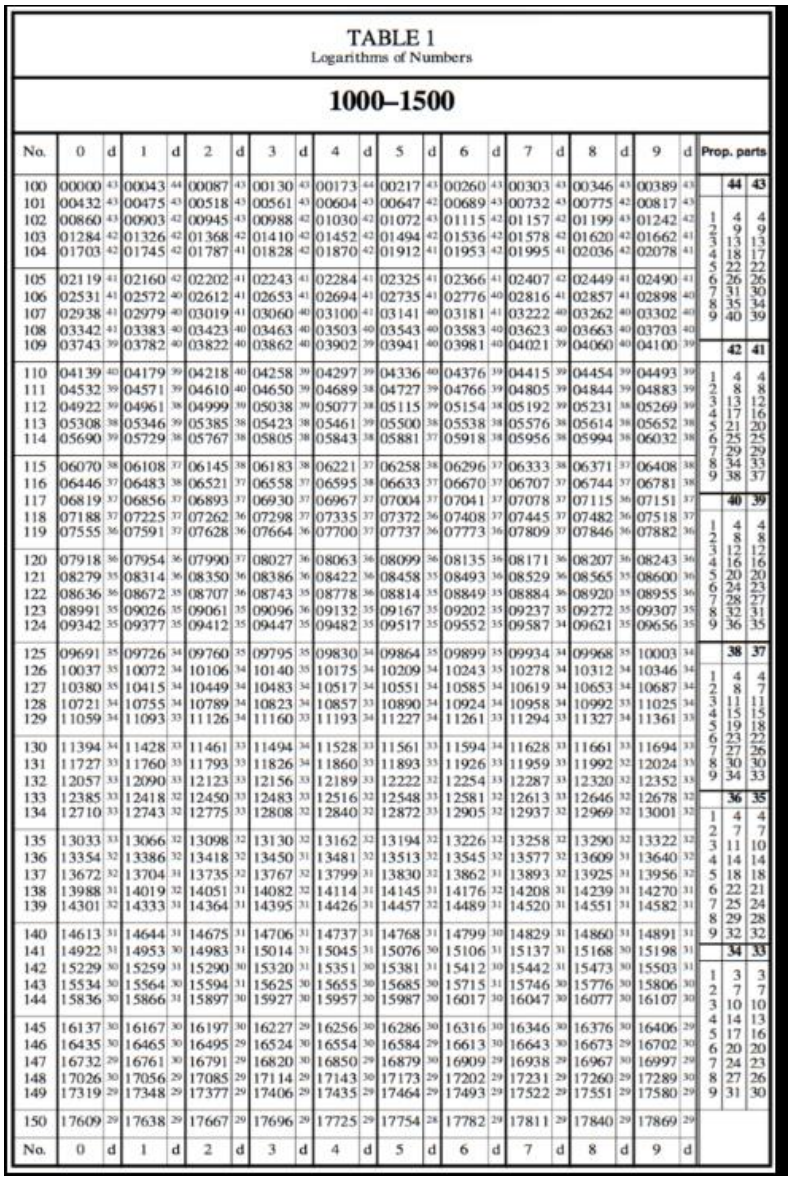


TABLE 1
Logarithms of Numbers

1000–1500

No.	0	d	1	d	2	d	3	d	4	d	5	d	6	d	7	d	8	d	9	d	Prop. parts
100	00000	43	00043	44	00087	43	00130	43	00173	44	00217	43	00260	43	00303	43	00346	43	00389	43	
101	00432	43	00475	43	00518	43	00561	43	00604	43	00647	42	00689	43	00732	43	00775	42	00817	43	
102	00860	43	00903	42	00945	43	00988	42	01030	42	01072	43	01115	42	01157	42	01199	43	01242	42	
103	01284	42	01326	42	01368	42	01410	42	01452	42	01494	42	01536	42	01578	42	01620	42	01662	41	
104	01703	42	01745	42	01787	41	01828	42	01870	42	01912	41	01953	42	01995	41	02036	42	02078	41	
105	02119	41	02160	42	02202	41	02243	41	02284	41	02325	41	02366	41	02407	42	02449	41	02490	41	
106	02531	41	02572	40	02612	41	02653	41	02694	41	02735	41	02776	40	02816	41	02857	41	02898	40	
107	02938	41	02979	40	03019	41	03060	40	03100	41	03141	40	03181	41	03222	40	03262	40	03302	40	
108	03342	41	03383	40	03423	40	03463	40	03503	40	03543	40	03583	40	03623	40	03663	40	03703	40	
109	03743	39	03782	40	03822	40	03862	40	03902	39	03941	40	03981	40	04021	39	04060	40	04100	39	
110	04139	40	04179	39	04218	40	04258	39	04297	39	04336	40	04376	39	04415	39	04454	39	04493	39	
111	04532	39	04571	39	04610	40	04650	39	04689	38	04727	39	04766	39	04805	39	04844	39	04883	39	
112	04922	39	04961	38	04999	39	05038	39	05077	38	05115	39	05154	38	05192	39	05231	38	05269	39	
113	05308	38	05346	39	05385	38	05423	38	05461	39	05500	38	05538	38	05576	38	05614	38	05652	38	
114	05690	39	05729	38	05767	38	05805	38	05843	38	05881	37	05918	38	05956	38	05994	38	06032	38	
115	06070	38	06108	37	06145	38	06183	38	06221	37	06258	38	06296	37	06333	38	06371	37	06408	38	
116	06446	37	06483	38	06521	37	06558	37	06595	38	06633	37	06670	37	06707	37	06744	37	06781	38	
117	06819	37	06856	37	06893	37	06930	37	06967	37	07004	37	07041	37	07078	37	07115	36	07151	37	
118	07188	37	07225	37	07262	36	07298	37	07335	37	07372	36	07408	37	07445	37	07482	36	07518	37	
119	07555	36	07591	37	07628	36	07664	36	07700	37	07737	36	07773	36	07809	37	07846	36	07882	36	
120	07918	36	07954	36	07990	37	08027	36	08063	36	08099	36	08135	36	08171	36	08207	36	08243	36	
121	08279	35	08314	36	08350	36	08386	36	08422	36	08458	35	08493	36	08529	36	08565	35	08600	36	
122	08636	36	08672	35	08707	36	08743	35	08778	36	08814	35	08849	35	08884	36	08920	35	08955	36	
123	08991	35	09026	35	09061	35	09096	36	09132	35	09167	35	09202	35	09237	35	09272	35	09307	35	
124	09342	35	09377	35	09412	35	09447	35	09482	35	09517	35	09552	35	09587	34	09621	35	09656	35	
125	09691	35	09726	34	09760	35	09795	35	09830	34	09864	35	09899	35	09934	34	09968	35	10003	34	
126	10037	35	10072	34	10106	34	10140	35	10175	34	10209	34	10243	35	10278	34	10312	34	10346	34	
127	10380	35	10415	34	10449	34	10483	34	10517	34	10551	34	10585	34	10619	34	10653	34	10687	34	
128	10721	34	10755	34	10789	34	10823	34	10857	33	10890	34	10924	34	10958	34	10992	33	11025	34	
129	11059	34	11093	33	11126	34	11160	33	11193	34	11227	34	11261	33	11294	33	11327	34	11361	33	
130	11394	34	11428	33	11461	33	11494	34	11528	33	11561	33	11594	34	11628	33	11661	33	11694	33	
131	11727	33	11760	33	11793	33	11826	34	11860	33	11893	33	11926	33	11959	33	11992	32	12024	33	
132	12057	33	12090	33	12123	33	12156	33	12189	33	12222	32	12254	33	12287	33	12320	32	12352	33	
133	12385	33	12418	32	12450	33	12483	33	12516	32	12548	33	12581	32	12613	33	12646	32	12678	32	
134	12710	33	12743	32	12775	33	12808	32	12840	32	12872	33	12905	32	12937	32	12969	32	13001	32	
135	13033	33	13066	32	13098	32	13130	32	13162	32	13194	32	13226	32	13258	32	13290	32	13322	32	
136	13354	32	13386	32	13418	32	13450	31	13481	32	13513	32	13545	32	13577	32	13609	31	13640	32	
137	13672	32	13704	31	13735	32	13767	32	13799	31	13830	32	13862	31	13893	32	13925	31	13956	32	
138	13988	31	14019	32	14051	31	14082	32	14114	31	14145	31	14176	32	14208	31	14239	31	14270	31	
139	14301	32	14333	31	14364	31	14395	31	14426	31	14457	32	14489	31	14520	31	14551	31	14582	31	
140	14613	31	14644	31	14675	31	14706	31	14737	31	14768	31	14799	30	14829	31	14860	31	14891	31	
141	14922	31	14953	30	14983	31	15014	31	15045	31	15076	30	15106	31	15137	31	15168	30	15198	31	
142	15229	30	15259	31	15290	30	15320	31	15351	30	15381	31	15412	30	15442	31	15473	30	15503	31	
143	15534	30	15564	30	15594	31	15625	30	15655	30	15685	30	15715	31	15746	30	15776	30	15806	30	
144	15836	30	15866	31	15897	30	15927	30	15957	30	15987	30	16017	30	16047	30	16077	30	16107	30	
145	16137	30	16167	30	16197	30	16227	29	16256	30	16286	30	16316	30	16346	30	16376	30	16406	29	
146	16435	30	16465	30	16495	29	16524	30	16554	30	16584	29	16613	30	16643	30	16673	29	16702	30	
147	16732	29	16761	30	16791	29	16820	30	16850	29	16879	30	16909	29	16938	29	16967	30	16997	29	
148	17026	30	17056	29	17085	29	17114	29	17143	30	17173	29	17202	29	17231	29	17260	29	17289	30	
149	17319	29	17348	29	17377	29	17406	29	17435	29	17464	29	17493	29	17522	29	17551	29	17580	29	
150	17609	29	17638	29	17667	29	17696	29	17725	29	17754	28	17782	29	17811	29	17840	29	17869	29	
No.	0	d	1	d	2	d	3	d	4	d	5	d	6	d	7	d	8	d	9	d	

An astronomer, Simon Newcomb, noticed that the pages at the beginning of the book were much more worn than those at the back of the book - which were hardly used at all – much like that shown in the picture of a well-thumbed book below.

Newcomb noticed that the leading digits, of all the numbers used in his calculations, were more likely to be small digits rather than large digits.

Newcomb published a note[2] about this and nothing more was heard about it – this was in1881.

Intuitively most people still felt that the digits 1 - 9 were evenly distributed in all numbers.

However, in 1937, a physicist by the name of Frank Benford, discovered Newcomb's idea and set about testing this idea using over 20,000 different sets of data such as: lengths of rivers, street addresses, death rates,

[1] https://www.wikiwand.com/en/Common_logarithm

[2] Newcomb, S. (1881). Note on the Frequency of Use of the Different Digits in Natural Numbers. *American Journal of Mathematics, 4*(1), 39-40.

sports statistics, molecular weights and so on – and it is base and scale invariant – the length of rivers could be in miles, kilometres, metres or even cubits.

Theory

Although you do not need to know the derivation[3] or proof of Benford's law, all you need to know is how to apply it to a set of data.

Benford's law states[4]:

$$\Pr(D_1 = d_1) = \log_{10}(1+\tfrac{1}{d_1}) \qquad d_1 = \{1,2,...,9\} \qquad \text{(Equation 1)}$$

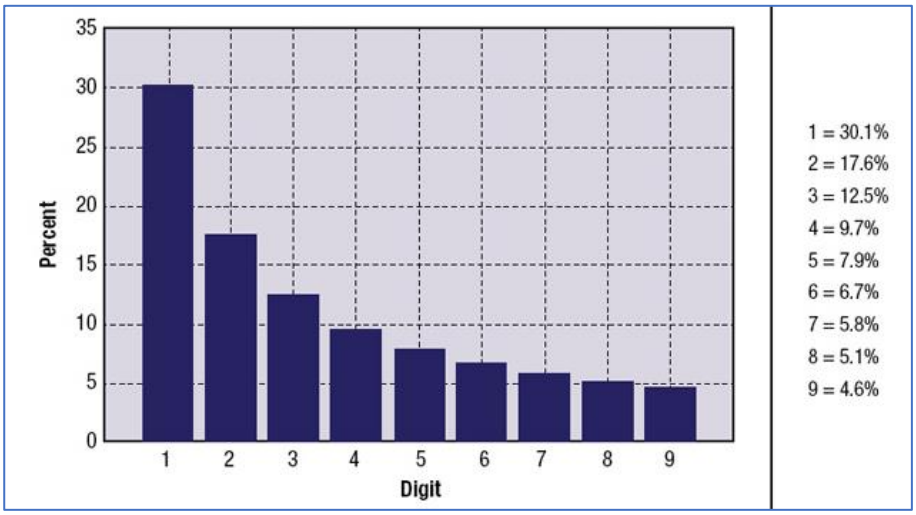
So that, for the first digit in a number, the probability that this digit is a '1' is:

$$\begin{aligned}\Pr(D_1 = 1) &= \log_{10}(1+\tfrac{1}{1}) \\ &= \log_{10}(2) \\ &= 0.3010\end{aligned}$$

or about 30.1%.

Similarly for the remaining digits 2-9.

If we do this for all the digits and plot them as a bar graph, we get:



[3] See, for example, Miller, S. J. (2015). A Quick Introduction to Benford's Law. In S. J. Miller (Ed.), *Benford's Law* (pp. 3-22): Princeton University Press.

[4] Nigrini, M. J., & ProQuest (Firm). (2012). *Benford's law applications for forensic accounting, auditing, and fraud detection, Wiley corporate F & A*, p.5

Your Task

Develop a Python program which will load up a set of data, determine the frequencies of the leading digits and compare them with the predicted distribution of Benford's law. Display this in a bar chart and a table of values. For example:

Digit 1:	Observed = 0.321	Expected = 0.301	
Digit 2:	Observed = 0.153	Expected = 0.176	and so on up till digit 9.

We shall look at three cases.

An Excel spreadsheet has been taken from Office-Watch: Benford's Law and Excel[5] to let you quickly visualize the Python application that we need make.

Case 1 - Fibonacci series[6]

This series begins with two numbers 1,1 – these two numbers are added to continue the series giving rise to the following (only the first 8 terms of the series are shown here):

```
1,1,2,3,5,8,13,21,. . .
```

There are many examples of this pattern in Nature and the series is closely related to the Golden[7] ratio.

Using the Excel spreadsheet generate a Fibonacci series up to the 24th term and see if the first digits obey Benford's Law. Does it get better if you add more terms?

The Chi-test[8] measures how close an actual value is to the expected value – the closer it is to 100% the closer the actual value is to the expected value. In our case, we are testing how close the frequency of each digit in our dataset is to Benford's prediction for that digit.

What is the value of the ChiTest comparison for this Fibonacci series? Does it get better if we add more terms to the series?

Case 2 – Fibonacci numbers & Benford's law using Python

In this case you are to repeat the analysis in Case 1 but using you Python code.

---

[5] https://office-watch.com/2012/benfords-law-and-excel/

[6] https://en.wikipedia.org/wiki/Fibonacci_number

[7] https://en.wikipedia.org/wiki/Golden_ratio

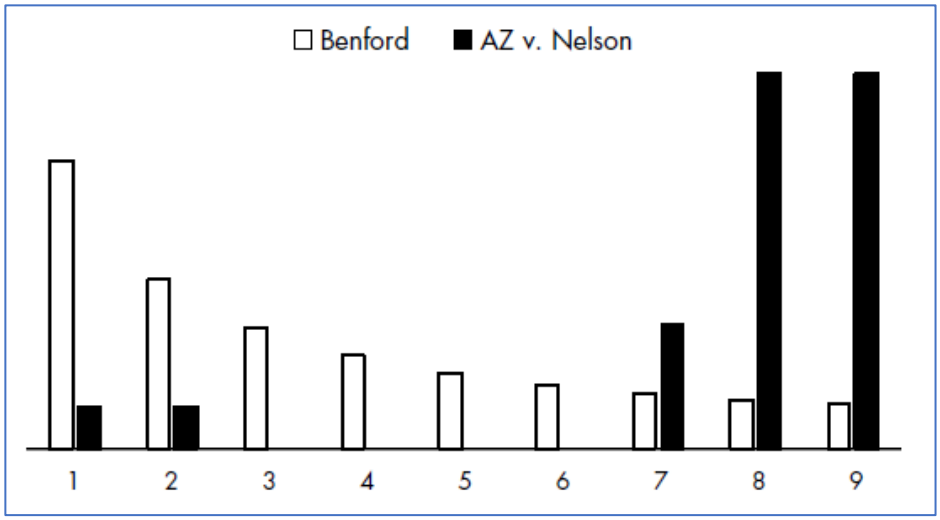
[8] Also written as χ^2-test

Case 3 – Length of Rivers[9] in the World

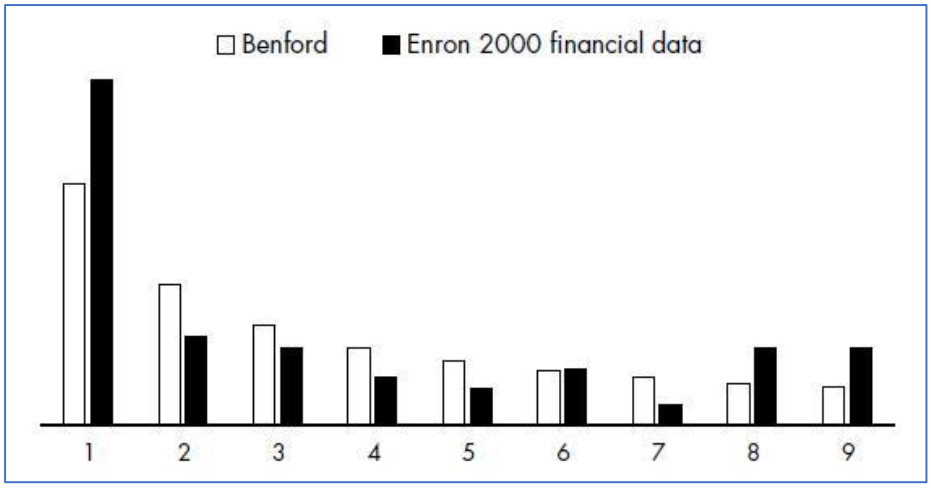
In this case, use your Python code to see whether the lengths of rivers in the world follow Benford's law.

Fraud detection using Benford's Law

One use of Benford's Law is to detect cases of Fraud. Consider the 1993 case of State of Arizona v Nelson. The accused diverted nearly $2M to fake vendors in an attempt to defraud the State. The frequency of first digits in the written cheques clearly violates Benford's Law leading to a conviction.



Another case is that of Enron in its posting of revenue for the year 2000. Comparison of the frequency of first digits versus the expected frequency shows large discrepancies. The company went bankrupt the following year – one of the greatest financial failures in history.



[9] https://en.wikipedia.org/wiki/List_of_rivers_by_length

Submission

A report is to be submitted in this assignment. There is a discussion section in the report in which you can apply step 6 in the six-step problem solving process and ask the four questions often used in evaluating a solution.

More details on academic reports are available - please refer to this link:

https://federation.edu.au/current-students/learning-and-study/online-help-with/guides-to-your-assessments

There are three important parts at the link above:

1. General Guide to Writing and Study Skills

This section describes the content of a report – refer to page 34 – Abstract, Table of Contents, Introduction and Conclusion and so on.

2. General Guide to Referencing

APA referencing style is described in this section – EndNote is also available to students

3. Assignment Layout and Appearance Guidelines

This section describes how the report should appear: margin sizes, fonts, how diagrams and tables are presented and so on.

You must supply your program source code files and your documentation, together **with any files** required to run your application, as a single zip file named as follows:

<YOUR-NAME>_<YOUR-STUDENT-ID>.zip

e.g. Ada_LOVELACE_30331815.zip

You may supply your word processed documentation in either Microsoft Word or LibreOffice/OpenOffice formats only – no proprietary Mac specific formats, please.

Assignments will be marked on the basis of fulfilment of the requirements and the quality of the work.

In addition to the marking criteria, marks may be deducted for failure to comply with the assignment requirements, including (but not limited to):

- Incomplete implementation(s), and
- Incomplete submissions (e.g. missing files), and
- Poor spelling and grammar.

You might be asked to demonstrate and explain your work.

Marking Criteria/Rubric

	Task	Mark
1	Pseudo-code for all Python scripts	10
2	Final Python code (Exceptions 2 marks), annotated with author details and with comments throughout the code (2 marks), consistent with pseudo-code	10
3	Tests to check that Python code is working correctly	10
4	Case 1 - Fibonacci numbers using example Excel sheet	5
5	Case 2 - Fibonacci numbers using your Python script – bar chart (10) & table (5)	15
6	Case 3 - Lengths of Rivers using your Python script – bar chart (10) & table (5)	15
7	Discussion (including 4 Questions in Step 6)	15
8	Report: Abstract, Title Page, Table of Contents (including Figures & Tables), Introduction, Method, Results, Discussion (including the 4 Questions in Step 6 of problem solving), Acknowledgements & Statement of Authorship, References	20
	TOTAL	100
	Final Grade	/20

Feedback

Ongoing feedback will be given in lectures and labs/tutes online classes and in arranged meeting. Feedback will also be given in Moodle.

Plagiarism

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found at http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism.