Frequency	**KA**	**DA**
deer:	[0.2692307692307692,	0.26471001263867433]
forest:	[0.04,	0.18313618441006976]
pangram:	[0.038461538461538464,	0.02857142857142857]
tree:	[0.34615384615384615,	0.4207882047009518]
woodm:	[0.22727272727272727,	0.19839142091152814]
1984:	[0.0,	0.0]
finnegan:	[0.16,	0.19766423357664234]

Textbook	**KA**	**DA**
deer:	[0.6923076923076923,	0.8811964611711838]
forest:	[0.0,	0.0]
pangram:	[0.0,	0.0]
tree:	[0.9230769230769231,	0.9933096013532278]
woodm:	[0.45454545454545453,	0.5817694369973191]
1984:	[0.0,	0.0]
finnegan:	[0.08,	0.049051094890510946]

Provided	**KA**	**DA**
deer:	[1.0,	1.0]
forest:	[0.11764705882352941,	0.08333333333333333]
pangram:	[0.46153846153846156,	0.5714285714285714]
tree:	[1.0,	1.0]
woodm:	[0.95,	0.9759036144578314]
1984:	[1.0,	1.0]
finnegan:	[0.9473684210526315,	0.9625]

Found	**KA**	**DA**
deer:	[1.0,	1.0]
forest:	[0.8,	0.9392174704276615]
pangram:	[0.23076923076923078,	0.3142857142857143]
tree:	[1.0,	1.0]
woodm:	[1.0,	1.0]
1984:	[0.125,	0.11538461538461539]
finnegan:	[1.0,	1.0]

1. The key used to encrypt the function does not affect the results. The letter that appear the most number of times in the text corresponds to “E”. The second most frequent letter will evaluate to “T” and no on. Since the key does not change the frequency of the characters in the text, manipulating the key does not change the results.
2. The data in provided text has much higher accuracy than the decipherment method using the frequency analysis in a5p1.py. The frequency analysis method is far more than perfect and does not have any “learn” methods in it.
3. Texts like “deer”, “tree” and “woodm” have one of the highest accuracies with “tree” being the highest. The biggest factor of such high accuracy is the length of the text. “Tree” has so many characters and for substitution cipher more data works the best. (NOTE: “forest” also has a high dataset, but it is in a different language – resulting in less accuracy)
4. We would use a function such as a .isEnglish() to check which sentences sound more like English words. If they don’t sound like English words, we can swap “E” with the “T” in the frequency analysis and repeat the process. This process may give us better results.

URL of the Solver - https://planetcalc.com/8047/