PHYC2050 Assignment #3, Winter 2023

Due midnight Wednesday Feb 8

Please submit a single file (jupyter-notebook, or a pdf with your code as snapshots/printout) that has your name in the file name. Remember that suggestions for how to solve a problem are suggestions – there are "many roads to Rome". Each problem is one mark. For each problem, include a test case or two to demonstrate that it is working, and some commentary.

1. **Distance in d dimensions**: Modify the KN routine crow() to calculate the euclidean distance (i.e. as the crow flies) between two points in arbitrary number of dimensions. [For this PS assume the two vectors are equal dimension, but you can "raise" an error if they aren't.]

2. **First digit distribution:** Write a function that measures the frequency of first non- zero digits of elements of an input array a. The output should be an array of frequencies (you can include '0' for simplicity, but you should get a frequency 0). To get the frequency count the number of times each digit is first, then divide each by the total number of elements. [There are many ways of doing this, but you could convert the number to a string, take the first character of the string, convert that to an integer, and use it as an index of your counting array.]

3. **First digits with random exponents:** Generate N = 100 000 random numbers of the form 10^x, where x is an unbiased random number in the range [−10,10]. Use your function from the previous question to evaluate the distribution of first digits of your random numbers. Compare this to Benford's Law (Google it). You should get very similar results for different ranges of x (as long as it is much wider range than [−1, 1]).

4. **Your own analytic function:** Using $e^x \simeq \sum_{n=0}^{N} \frac{x^n}{n!}$, write your own exponential function my_exp(x,N). [Here N is the maximum power you evaluate in the sum. You can use a default N = 100 in your definition.]

5. **Errors:** Comparing my_exp(x,N) and np.exp(x), plot the absolute error of your function vs x for N = 1, 10, 100. Also plot the absolute relative errors. [Where |f 1 − f 2| is absolute error, and |f 1/f 2 − 1| is absolute relative error between two functions.] Pick your limits and ranges and linear/log scales to illustrate where my_exp() works well (and where it doesn't)