7MR10020: Scientific Programming
Assignment 1

Some of the experiments run (ST = sample time, SR = sample rate, c = cosine, r = random):

No.	Experiment	Pearson's Coeff	Fourrier Product	DTW
1	c vs c: $ST_1 = ST_2$ and $SR_1 = SR_2$	1.000	1.000	1.000
2	c vs c: $ST_1 \neq ST_2$ and $SR_1 = SR_2$	~0.000	1.000	0.889
3	c vs c: $ST_1 = ST_2$ and $SR_1 \neq SR_2$	1.000	1.000	0.925
4	c vs c: $ST_1 \neq ST_2$ and $SR_1 \neq SR_2$	~0.000	1.000	0.831
5	c vs c: $ST_1 = 2 * ST_2$ and $SR_1 = SR_2$	~0.000	1.000	0.833
6	c vs c: $ST_1 = ST_2$ and $SR_1 = SR_2$ (with random noise; run multiple times)	0.925avg	0.992avg	0.878avg
7	r vs r: $ST_1 \neq ST_2$ and $SR_1 \neq SR_2$	0.0317avg	0.0486avg	0.823avg
8	r vs r: $ST_1 \neq ST_2$ and $SR_1 = SR_2$	0.0420avg	0.0249avg	0.853avg
9	r vs r: $ST_1 = ST_2$ and $SR_1 \neq SR_2$	-0.235avg	0.114avg	0.826avg

More focus was given to cosine signals as their results are easier to reproduce and the effect of changing the inputs is more obvious. The exact parameters of these can be found in "Main.py"

Pearson's Correlation Coefficient (PCC):

PCC doesn't seem to capture the difference between signals that well in our case. PCC captures differences only in sample time and so a different sampling rate doesn't affect our results. Using Experiment 5 as an example, the two signals should be similar, but PCC returns that they are 100% decorrelated. One of the good things about PCC is the similarity can be negative, which shows us the direction of the similarity.

Fourier transform of the product of two signals:

Fourier is probably the least fitted measure to capture the difference of cosine signals. This is because it is a frequency domain measurement and cosine signals have the same frequency.

Distance Time Warping (DTW):

The important part about distance time warping is that it standardises time. An example where this might be relevant is in the case of speech. If a person talks quickly or slowly, using a simple Euclidean distance measure would determine that a different person is speaking. By eliminating the effect of time, DTW would find that the two voice signals are similar. In our case, comparing two cosine signals with different sample times returns that they are similar but not 100% similar which makes sense (experiments 2, 4, and 5). In addition to that, keeping the sample time the equal between the two signals and changing only the sampling rate affects our results (Experiment 3).

Best Measure:

In general it was hard to evaluate random signals and so most of the focus was given to cosine signals. Therefore, this was a close call between PCC and DTW. In the end, I've chosen DTW as the best measure as it considers changes to both the sampling time and sampling rate, returning a high similarity (in the case of cosine signals) which indeed shows that two different cosine signals are similar even when these parameters are different. In addition to that, DTW doesn't require us to resample (up/down) our signals which means that we can compare the original signals. Moreover, DTW handles the simulated noise I added to the signals better than all three methods.