Code	SUPPORT_VECTOR_MACHINE.PY
Author	Nathaniel Heatwole, PhD (heatwolen@gmail.com) (GitHub) (LinkedIn)
Summary	Uses linear support vector machine (SVM) to separate two groups of data, both from scratch and using sklearn, and also fits quadratic and cubic SVMs (using sklearn)

Methods/ Process	Support vector machine
	- Supervised learning method for generating boundary between groups of data
	- Classification threshold (hyperplane) can be linear or non-linear
	- *Support vectors* are the *margin* boundaries, and have the same functional form and slope as and are equidistant about the boundary vector
	- Optimization:
	    - *Hard margin*: no points permitted in region between support vectors ("demilitarized zone," so to speak), and margin width maximized
	    - *Soft margin*: number of points inside margin and/or their distance inside is minimized (for overlapping groups)
	
	Steps (from scratch)
	1. Randomly select point from each group in the training data
	2. Take negative inverse of slope of line connecting those points – this new vector is orthogonal to line connecting the points, and is therefore an efficient initial guess for the SVM slope[1]
	3. Fit line with this slope through each point in training data
	4. Identify support vectors, using group-level extremities of y-intercepts (from previous step), and considering the relative orientation of the groups
	5. Compute the margin (distance) between the two support vectors
	6. Optimize slope value (from step #2) to find local maximum margin width, using parametric variation (fixed number of iterations)
	7. Repeat this entire process for many different sets of randomly selected points
	8. Choose the SVM parameter set that yields the best separation (global maximum margin)

Training Data	Synthetic data (randomly generated) consisting of two linearly separable groups
Results	Two methods align well (from scratch and using sklearn)

[1] This is because, rather than *connecting* the data, as linear fits do, SVMs best *separate* the groups.