1. Write a function *angle\_between(v1, v2)* where v1 and v2 are two vectors that are passed in, and the angle between them is calculated.
2. Write a function called row\_dot(A, r1, r2) which takes an input matrix called A, one row identified by r1, and another row identified by r2 and returns the dot product of the r1 and r2 rows (indexing starts at 0).
3. Write a function matrix\_division(m1, m2) that takes in two matrices, m1 and m2, and returns the result. What is the trick with matrix division?
4. Write a function is\_orthogonal(v1,v2, tol), where v1 and v2 are column vectors of the same size and tol is a scalar value strictly larger than 0. The output should be 1 if the angle between v1 and v2 is within tol of π/2; that is, |π/2−θ|<tol|π/2−θ|<tol, and 0 otherwise. You may assume that v1 and v2 are column vectors of the same size, and that tol is a positive scalar.

*# Test cases for problem 4*

a = np.array([[1], [0.001]])

b = np.array([[0.001], [1]])

*# output: 1*

is\_orthogonal(a,b, 0.01)

*# output: 0*

is\_orthogonal(a,b, 0.001)

*# output: 0*

a = np.array([[1], [0.001]])

b = np.array([[1], [1]])

is\_orthogonal(a,b, 0.01)

*# output: 1*

a = np.array([[1], [1]])

b = np.array([[-1], [1]])

is\_orthogonal(a,b, 1e-10)

1. Create a class called vector\_calculator. \_\_init\_\_ should take self, vector1, and vector2. It should have two methods inside of it that are from problems 1 and 4 above (*angle\_between and is\_orthogonal)*. Make sure tol is defaulted but can be overwritten. Make sure you can call angle\_between and is\_orthogonal through the class and return the correct results as in problems 1 and 4.
2. Show a graphical representation of the breast cancer data outlined in this reading <https://www.datacamp.com/community/tutorials/principal-component-analysis-in-python> Be sure to describe what PCA is and incorporate the principal components in your plot.