EAS 596, Spring 2020, Final Exam, Part II
Due 11:59 PM Sunday May 17, 2020, submitted to UBlearns and UBBox.
Total Possible Points: 30

Work **all** problems. Please read all directions carefully. You may use course notes, books, internet search, and tools such as Matlab or Python. If you use external references such as journal papers, books, or internet search results, be sure to cite them accordingly. **Consultation, either in-person or virtually, with anyone other than Dr. Salac is STRICTLY FORBIDDEN.** All work submitted must be your own! Make sure you code is well organized. Points will be deducted if the logic behind your code is not apparent.

All required files are on the shared UBBox folder. Submit all code to your UBBox folder.

I affirm that I will not give or receive any unauthorized help on this exam and that all work will be my own. I will complete this exam in a fair, honest, respectful, responsible, and trustworthy manner. This means that I will complete the exam as if the professor was watching my every action. I will act according to the professors instructions, and I will neither give nor receive any aid or assistance other that what is authorized. I know that the integrity of this exam and this class is up to me, and I pledge not to take any action that would break the trust of my classmates or professor, or undermine the fairness of this class.

Problem 1 (15 pts.)

This problem will use a simple Monte Carlo method to determine the mean and variance of the deflection and angle of a simple circular cantilever beam.

You have been provided a MATLAB function `[deflect, angle] = eas596_final_beamDeflection(F, E, d)`, which takes as input the force at the tip, `F`, the elastic modulus, `E`, and the diameter of the beam, `d`. It returns the deflection of the tip, `deflect`, and the angle of deflection of the tip, `angle`.

Assume that the force, elastic modulus, and diameter all obey Gaussian distributions with the following means and standard deviation:

	μ	σ
F (N)	1000	10
E (Pa)	200×10^9	2.5×10^9
d (m)	10^{-2}	5×10^{-4}

Create a MATLAB function `[mu, V] = ubitname_final_p1(n)` which returns a 2×1 vector `mu` which has the sample mean for deflection in location 1 and that for the angle in location 2. The matrix `V` is the covariance matrix between the deflection and angle. The input `n` is the number of random samples to draw.

You are allowed to use the MATLAB commands `sum`, `zeros`, `normrnd`, along with for-loops, while-loops, etc. If you wish to use any other MATLAB function check with the instructor before using it. Review the help-files for information on `normrnd` if you are not familiar with it. Note that `eas596_final_beamDeflection` allows for scalar or vector inputs.

On the UBLearns Part-2 Exam site submit the means and covariances for `n=1e6`.

Problem 2 (15 pts)
This problem will use Principal Component Analysis to determine outliers from a given data set.

The file `CityData.mat` contains ratings for nine categories for 250 cities. The file has three pieces of information:

- `names`: A length-250 vector containing the anonymized city IDs, from 1 to 250.
- `categories`: A length-9 cell-array listing the nine categories each city is rated on.
- `ratings`: A 250×9 matrix with the ratings for each city.

Create a MATLAB function `[scores, L] = ubitname_final_p2()` which does the following:

1. Load the data using `load('CityData.mat');`
2. Performs the principal component analysis (PCA) on `ratings`. Note that this has 9-data points of interest and 250-observations/samples.
3. Returns the principal magnitude (eigenvalues) in `L` as an ordered-vector from largest to smallest.
4. Determines the PCA scores for all principal directions and returns it in `scores`.

Before performing the PCA be sure the shift each data set so that it has zero mean. You are allowed to use the following MATLAB functions: `zeros`, `sum`, `eigs`, along with for-loops, while-loops, etc. If you wish to use any other MATLAB function check with the instructor before using it.

On the UBLearns Part-2 Exam site submit the numeric ID of the city which is clearly an outlier. *NOTE: You do not need to submit any code you have written to determine the outlier from the scores obtained using* `ubitname_final_p2`.