CS 491/691 Project 4
Due 12/19 11:59pm

Logistics: Logistics: You must implement everything stated in this project description that is marked with an **implement** tag. Whenever you see the **write-up** tag, that is something that must be addressed in the write-up for the project. **You may only use numpy and matplotlib.pyplot. No late work will be accepted.** I advise that you submit to Webcampus regularly and well before the deadline.

Deliverables: Each student should submit a single ZIP file, containing your project code (PCA.py) and your writeup (PDF with source files, or README.txt). Your zip file should be named lastname_firstname_project4.zip. Your code should run without errors on the ECC linux machines. If your code does not run for a particular problem, you will lose 50% on that problem. You should submit only one py file, named accordingly. A test_script.py file is provided with two arrays of test data.

Extra Credit: Undergraduate students who use LaTeX for their write-up and submit their LaTeX source files will get up to 5 extra points.

File name: PCA.py

1 Covariance Matrix (25 points)

Implement: You will implement the following function

```
def compute_covariance_matrix(Z)
```
The above function will take the data matrix `Z` and return the covariance matrix $Z^T Z$=`COV` (a numpy array).

2 PCS (25 points)

Implement: You will implement the following function

```
def find_pcs(cov)
```
The above function will take the covariance matrix `cov` and return the ordered (largest to smallest) principal components `PCS` (a numpy array where each column is an eigenvector) and corresponding eigenvalues `L` (a numpy array). You will want to use `np.linalg.eig` for this.

3 Project Data (25 points)

Implement: You will implement the following function

```
def project_data(Z, PCS, L)
```
The above function will take the data matrix `Z`, the principal components `PCS`, and corresponding eigenvalues `L`. We expect you to maintain 1 principal component (project all the data onto a single axis). This function will return `Z_star`, the projected data.

4 Plotting (25 points)

Implement: You will implement the following function

```
def show_plot(Z, Z_star)
```
The above function will take the original data `Z`, and the projected data `Z_star`. On the same figure, plot the original data `Z` as a scatter plot, and the projected data `Z_star` as a line. Save this data into an image file (.jpg preferred) and turn it in with the report. For the report give a brief explanation of the benefit to principal component analysis when dealing with higher dimensional feature vectors.