

For this lab, you will learn to write a MATLAB function. This will be a simple function with several forms, so you get to do input/output argument checking and implement the tasks accordingly.

This function is called `randn2d` and does the data generation of Assignment #2. Your implementation should be able to handle the following combinations of input and output arguments:

1. `X = randn2d(n)`      % **X** is an **nx2** array of the generated samples.  
                             % **n** is the number of generated samples (>0).  
                             % The distribution has zero mean and identity covariance matrix.
2. `X = randn2d(n,s,a,u)` % **s** is a 2-element vector representing the scaling factor (standard deviation)  
                             % of the two axes of the 2-D normal distribution.  
                             % You have to ensure that **s** contains only positive values.  
                             % **a** is the rotation angle (counter-clockwise) of the major axis from the x axis.  
                             % **u** (a 2-element vector) is the displacement of the center of the distribution.
3. `X = randn2d(n,C,u)`    % **C** is the covariance matrix used to generate the sample.  
                             % You have to ensure that **C** is symmetric and has all-positive eigenvalues.  
                             % The major and minor axes of the distribution are the eigenvectors of **C**.  
                             % The scaling factors are the square roots of the eigenvalues of **C**.
4. `X = randn2d(_, 'plot')`    % If the last input is the **char** vector **'plot'** beyond the previous forms,  
                             % plot the samples in a figure.  
                             % You can use **ischar** to **strcmpi** to check  
                             % whether the last argument is **'plot'**.
5. `[X,Ct,ut] = randn2d(_)`    % **Ct** and **ut** are the estimated covariance matrix and mean of **X**  
                             % Compute them only if three output arguments are requested.

#### Notes:

- In the specs above, we use **'\_'** in the argument list to represent "all previous combinations". This is quite common in the documentation.
- You need to do input argument checking. Try to utilize **validateattributes** when possible. Your program needs to send out meaningful error messages if the numbers/types/values of input or output arguments are invalid.