2D Confidence Ellipses

from http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/

error_ellipse.m

```
clear all;
close all;

% Create some random data
s = [2 2];
x = randn(334,1);
y1 = normrnd(s(1).*x,1);
y2 = normrnd(s(2).*x,1);
data = [y1 y2];

% Calculate the eigenvectors and eigenvalues
covariance = cov(data);
[eigenvec, eigenval ] = eig(covariance);

% Get the index of the largest eigenvector
[largest_eigenvec_ind_c, r] = find(eigenval == max(max(eigenval)));
largest_eigenvec = eigenvec(:, largest_eigenvec_ind_c);

% Get the largest eigenvalue
largest_eigenval = max(max(eigenval));

% Get the smallest eigenvector and eigenvalue
if(largest_eigenvec_ind_c == 1)
    smallest_eigenval = max(eigenval(:,2))
    smallest_eigenvec = eigenvec(:,2);
else
    smallest_eigenval = max(eigenval(:,1))
    smallest_eigenvec = eigenvec(1,:);
end

% Calculate the angle between the x-axis and the largest eigenvector
angle = atan2(largest_eigenvec(2), largest_eigenvec(1));

% This angle is between -pi and pi.
% Let's shift it such that the angle is between 0 and 2pi
if(angle < 0)
    angle = angle + 2*pi;
end

% Get the coordinates of the data mean
avg = mean(data);

% Get the 95% confidence interval error ellipse
```

```
chisquare_val = 2.4477;
theta_grid = linspace(0,2*pi);
phi = angle;
X0=avg(1);
Y0=avg(2);
a=chisquare_val*sqrt(largest_eigenval);
b=chisquare_val*sqrt(smallest_eigenval);

% the ellipse in x and y coordinates
ellipse_x_r  = a*cos( theta_grid );
ellipse_y_r  = b*sin( theta_grid );

%Define a rotation matrix
R = [ cos(phi) sin(phi); -sin(phi) cos(phi) ];

%let's rotate the ellipse to some angle phi
r_ellipse = [ellipse_x_r;ellipse_y_r]' * R;

% Draw the error ellipse
plot(r_ellipse(:,1) + X0,r_ellipse(:,2) + Y0,'-')
hold on;

% Plot the original data
plot(data(:,1), data(:,2), '.');
mindata = min(min(data));
maxdata = max(max(data));
xlim([mindata-3, maxdata+3]);
ylim([mindata-3, maxdata+3]);
hold on;

% Plot the eigenvectors
quiver(X0, Y0, largest_eigenvec(1)*sqrt(largest_eigenval),
largest_eigenvec(2)*sqrt(largest_eigenval), '-m', 'LineWidth',2);
quiver(X0, Y0, smallest_eigenvec(1)*sqrt(smallest_eigenval),
smallest_eigenvec(2)*sqrt(smallest_eigenval), '-g', 'LineWidth',2);
hold on;

% Set the axis labels
hXLabel = xlabel('x');
hYLabel = ylabel('y');
```