0.0.1 Question 1.3

M is a 2×2 real-valued symmetric matrix with eigenvalues $\lambda_1 = 2$, $\lambda_2 = -1$ and corresponding eigenvectors

$$u_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}, u_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

What are the eigenvalues of the matrix $M + 2I$ (where I is the identity matrix)?

Please prove this or hand calculate it, don't use libraries to do it for you numerically.

Points: 0.3

$$\begin{aligned}
&(M + 2I)x = Mx + 2Ix = \lambda x + 2x = (\lambda + 2)x \\
&\lambda_1 = 2 + 2 = 4 \\
&\lambda_2 = -1 + 2 = 1
\end{aligned}$$

0.0.2 Question 1.4

M is a 2×2 real-valued symmetric matrix with eigenvalues $\lambda_1 = 2$, $\lambda_2 = -1$ and corresponding eigenvectors

$$u_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}, u_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

What are the eigenvalues of the matrix $M^2 = MM$?

Please prove this or hand calculate it, don't use libraries to do it for you numerically.

Points: 0.3

$$\begin{aligned}
&M^2 x = M(Mx) = M(\lambda x) = \lambda(Mx) = \lambda^2 x, \\
&\lambda_3 = 2^2 = 4, \\
&\lambda_4 = (-1)^2 = 1, \\
&\text{where } \lambda_3 \text{ and } \lambda_4 \text{ are the eigenvalues of } M^2.
\end{aligned}$$

0.0.3 Question 1.5

If $\mathbf{v}$ is an eigenvector of the matrix $A^T A$ with eigenvalue λ, show that $A\mathbf{v}$ is an eigenvector of AA^T with the same eigenvalue λ.

Remember to obey the rules for *matrix* multiplication.

BTW, this is a cool linear algebra trick we will use later to make calculations easier when A has a huge number of rows and a reasonable number of columns.

Points: 0.8

$$A^T Av = \lambda v AA^T(Av) = A(\lambda v) = \lambda(Av)$$

0.0.4 Q3.4 Interpret Eigenvalues and Eigenvectors

Please explain what these eigenvectors represent, and what the associated eigenvalues represent as well.

Points: 0.3

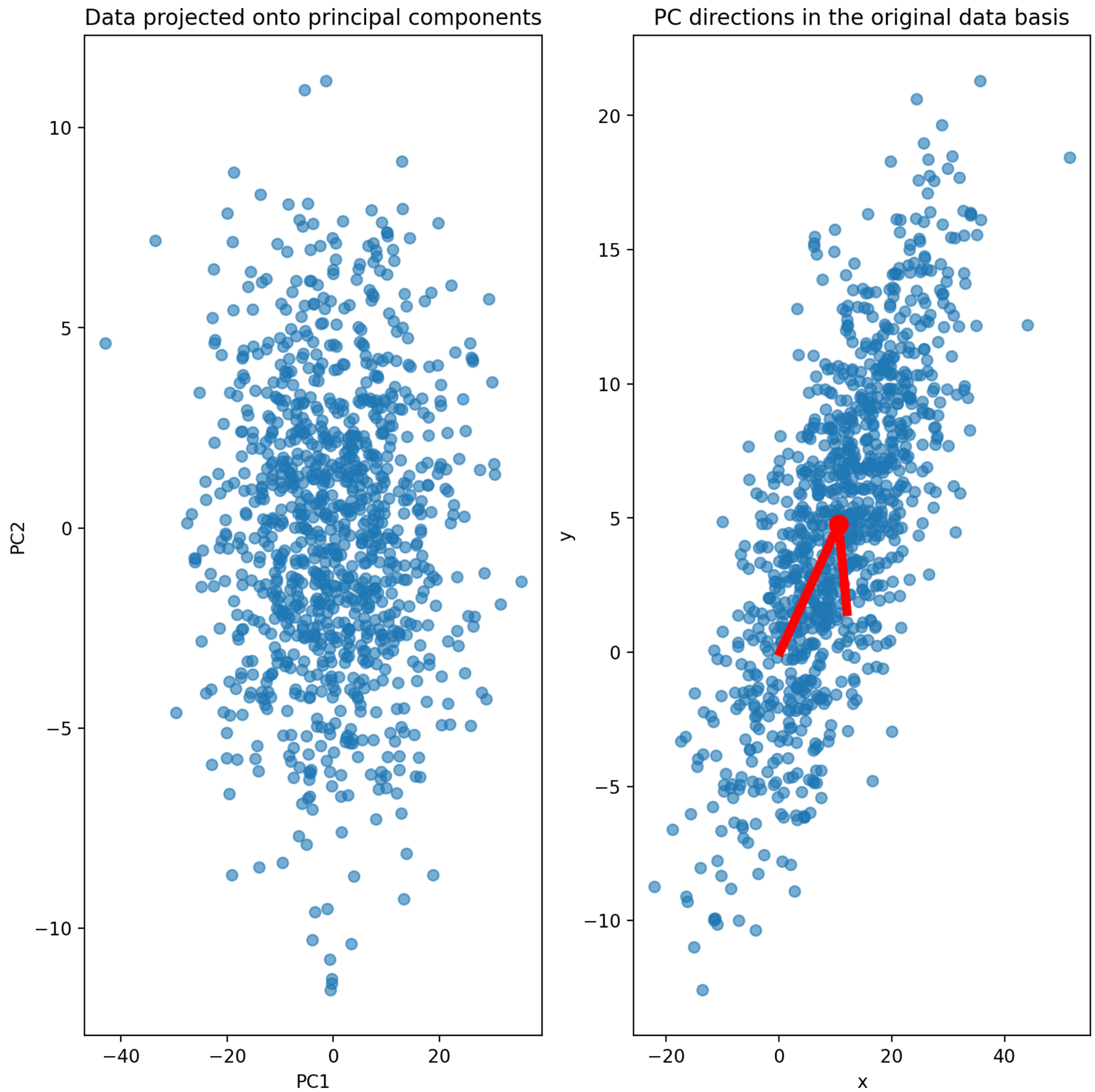
The eigenvectors of the covariance matrix Σ represent the principal components of the data (where the first PC is the direction that maximises variance in the data, and subsequent ones are the direction of maximum variance under the constraint of orthogonality), and their corresponding eigenvalues represent the amount of variability in the data along that eigenvector's dimension.

0.0.5 Q3.5 PCA Projection and Plot

1. PLOT #1: You will project the mean centered data onto its principal component basis. Make a scatter plot of the projected data. The graph should have a title ("Data projected onto principal components") and axis labels ("PC1", "PC2").
2. PLOT #2: You will visualize the PCs in the data's original coordinates. The graph should have a title ("PC directions in the original data basis") and axis labels ("x", "y"). Lay to following plots on top of each other to form a single graph:
   - Scatter plot the original data
   - Plot a large red dot (use `c='r', s=100`) at the mean of the data
   - using commands of the form `plt.plot([vec_x_start, vec_x_end],[vec_y_start, vec_y_end], c='r', linewidth=5)` plot both PC1 and PC2 in wide, red lines. Note that
     - PCs should start at the mean of the data
     - PCs should be scaled by the square root of their eigenvalues (remember we squared the data to get the covariance matrix)

Points: 0.2

```
In [60]: # plot those two graphs here.

         proj = eigenvectors.T.dot(Z)
         fig, axs = plt.subplots(1,2, figsize =(10,10))
         axs[0].scatter(x=proj[0,:], y=proj[1,:], alpha=0.6)
         axs[0].set_title('Data projected onto principal components')
         axs[0].set_xlabel('PC1')
         axs[0].set_ylabel('PC2')


         axs[1].scatter(x=X[0,:], y=X[1,:], alpha=0.6)
         axs[1].scatter(mu[0],mu[1], c='r', s=100)
         mu_flat = mu.flatten()

         pc1_end = mu_flat + eigenvectors[:, 0] * np.sqrt(eigenvalues[0])
         pc2_end = mu_flat + eigenvectors[:, 1] * np.sqrt(eigenvalues[1])

         axs[1].plot([mu_flat[0], pc1_end[0]], [mu_flat[1], pc1_end[1]], 'r-', linewidth=5)
         axs[1].plot([mu_flat[0], pc2_end[0]], [mu_flat[1], pc2_end[1]], 'r-', linewidth=5)
         axs[1].set_title('PC directions in the original data basis')
         axs[1].set_xlabel('x')
         axs[1].set_ylabel('y')

         plt.show();
```


Data projected onto principal components
PC1
PC2
−40
−20
0
20
10
5
0
−5
−10
PC directions in the original data basis
x
y
−20
0
20
40
20
15
10
5
0
−5
−10

0.0.6 Question 4.6 Non-face Image Projection

Does the reconstructed dog look like the original picture? Explain why the reconstruction looks the way it does.

Points: 0.4

The reconstructed dog is very different from the original picture. This is because, to reconstruct the image, we used the eigenfaces calculated on the `faces` dataset (matrix of eigenvectors of AA^T) and the average pixel values of a face. To reconstruct a more accurate dog image we should have calculated the eigenvectors and mean image from the dog dataset.