ISYE 6740 - Summer 2024

Homework 2

Parth Patel

1 - Conceptual questions

Question 1.1

The first principle component direction v corresponds to the largest eigenvector of the sample covariance matrix:

$$v = \arg\max_{w:\|w\|\leq 1} \frac{1}{m}\sum_{i=1}^{m}(w^T x^i - w^T\mu)^2.$$

Proof using steps from Module 4 - PCA Dimensionality Reduction lecture:

Given m data points $\{x^1, x^2, \ldots, x^m\} \in \mathbb{R}^n$, we want to find the direction w that maximizes the variance of the data projected onto w.

The mean of the data can be defined as:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$$

The variance of the data projected onto w is:

$$\text{Var} = \frac{1}{m}\sum_{i=1}^{m}(w^T x^i - w^T\mu)^2 = \frac{1}{m}\sum_{i=1}^{m}(w^T(x^i-\mu))^2$$

where the goal is to maximize the equation above subject to $\|w\| \leq 1$.

Simplifying the terms further:

$$\frac{1}{m}\sum_{i=1}^{m} w^T(x^i-\mu)(x^i-\mu)^T w$$

The sample covariance matrix C can be defined as:

$$C = \frac{1}{m}\sum_{i=1}^{m}(x^i-\mu)(x^i-\mu)^T$$

This gives the following objective function:

$$\frac{1}{m}\sum_{i=1}^{m} w^T(x^i-\mu)(x^i-\mu)^T w = w^T C w$$

$$\max_{w:\|w\|\leq 1} w^T C w$$

To solve the constrained optimization problem, I can use the method of Lagrange multipliers. The Lagrangian can be defined as:

$$L(w,\lambda) = w^T C w + \lambda(1-\|w\|^2)$$

If w is a maximum of the original optimization problem, then there must exist a λ where (w,λ) is a stationary point of $L(w,\lambda)$.

By taking the partial derivative of the Lagrangian with respect to w and setting it to zero, we can see that:

$$\frac{\partial L}{\partial w} = 0 = 2Cw - 2\lambda w = 0 \implies Cw = \lambda w$$

This shows that the optimal solution w must be an eigenvector of C. Since the objective function becomes λ that is associated with w, w should be the eigenvector corresponding to the largest eigenvalue λ_1 in order to maximize $w^T C w$.

Therefore, the direction v that maximizes the variance of the projected data is the eigenvector of the covariance matrix C corresponding to the largest eigenvalue. This eigenvector is the first principal component direction.

Question 1.2

Based on the outline given in the lecture, it can be show that the maximum likelihood estimate (MLE) for Gaussian random variable using observations $x^1,\dots,x^m$, that are *i.i.d.* (independent and identically distributed) following the distribution $\mathcal{N}(\mu,\sigma^2)$, and the mean and variance parameters are given by

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^i, \quad \hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i-\hat{\mu})^2,$$

respectively.

Proof using online sources (StatLect, StatQuest YouTube, & Mike the Mathematician YouTube) and the outline given in the lectures:

The probability density function of a Gaussian random variable x with mean μ and variance σ^2 is:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Given m i.i.d. observations $x^1, x^2, \ldots, x^m$, the likelihood function $L(\mu, \sigma^2)$ is the product of the individual densities:

$$L(\mu, \sigma^2) = \prod_{i=1}^{m} f(x^i|\mu, \sigma^2) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x^i - \mu)^2}{2\sigma^2}\right)$$

To simplify calculations, I can take the natural logarithm of the likelihood function, giving me the log-likelihood function $\ell(\mu, \sigma^2)$:

$$\ell(\mu, \sigma^2) = \ln L(\mu, \sigma^2) = \sum_{i=1}^{m} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x^i - \mu)^2}{2\sigma^2}\right)\right)$$

$$\ell(\mu, \sigma^2) = \sum_{i=1}^{m} \left(-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x^i - \mu)^2}{2\sigma^2}\right)$$

$$\ell(\mu, \sigma^2) = -\frac{m}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(x^i - \mu)^2$$

To find the values of μ and σ^2 that maximize $\ell(\mu, \sigma^2)$, I can take the partial derivatives with respect to μ and σ^2, set them to zero, and solve for the parameters.

The partial derivative of $\ell(\mu, \sigma^2)$ with respect to μ:

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \mu} = -\frac{1}{2\sigma^2}\frac{\partial}{\partial \mu}\sum_{i=1}^{m}(x^i - \mu)^2$$

$$\frac{\partial}{\partial \mu}\sum_{i=1}^{m}(x^i - \mu)^2 = \sum_{i=1}^{m} 2(x^i - \mu)(-1) = -2\sum_{i=1}^{m}(x^i - \mu)$$

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{m}(x^i - \mu)$$

Setting the derivative to zero and solving for μ:

$$\frac{1}{\sigma^2}\sum_{i=1}^{m}(x^i - \mu) = 0$$

$$\sum_{i=1}^{m} x^i - m\mu = 0$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^i = \hat{\mu}$$

Next, I can compute the partial derivative of $\ell(\mu, \sigma^2)$ with respect to σ^2:

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \sigma^2} = -\frac{m}{2}\frac{1}{\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{m}(x^i - \mu)^2$$

Setting the derivative to zero:

$$-\frac{m}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{m}(x^i - \mu)^2 = 0$$

Multiplying both sides by $2(\sigma^2)^2$:

$$-m\sigma^2 + \sum_{i=1}^{m}(x^i - \mu)^2 = 0$$

Solving for σ^2:

$$m\sigma^2 = \sum_{i=1}^{m}(x^i - \mu)^2$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \mu)^2 = \hat{\sigma}^2$$

Therefore, the maximum likelihood estimates for the parameters of the Gaussian distribution are:

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^i, \quad \hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \hat{\mu})^2$$

These results match the expressions for the sample mean and sample variance.

Question 1.3

The ISOMAP (Isometric Mapping) algorithm, a method for manifold learning and nonlinear dimensionality reduction, is built on three key ideas:

1. ISOMAP starts by constructing a neighborhood graph to capture the local geometry of the data. This involves determining which points are neighbors on the manifold. Two common methods are connecting each point to all points within a fixed radius ϵ and connecting each point to its K nearest neighbors. These neighborhood relations are represented in an adjacency matrix A with entries recording the Euclidean distances between neighboring points.
2. ISOMAP estimates the geodesic distances between all pairs of points on the manifold. This is done by computing the shortest path distances based on the adjacency matrix A. The geodesic distance between two points is approximated by the sum of the shortest paths connecting them through their neighbors which captures the intrinsic geometry of the manifold. These shortest paths can be computed using the Floyd-Warshall algorithm or Dijkstra's algorithm.
3. In the final step, ISOMAP applies Multidimensional Scaling (MDS) to the matrix of graph distances. This involves finding a low-dimensional embedding of the data that best preserves the manifold's intrinsic geometry. The goal is to place the data points in a low-dimensional space such that the pairwise distances in this space approximate the geodesic distances on the manifold. This is done by minimizing a cost function through eigendecomposition.

Question 1.4

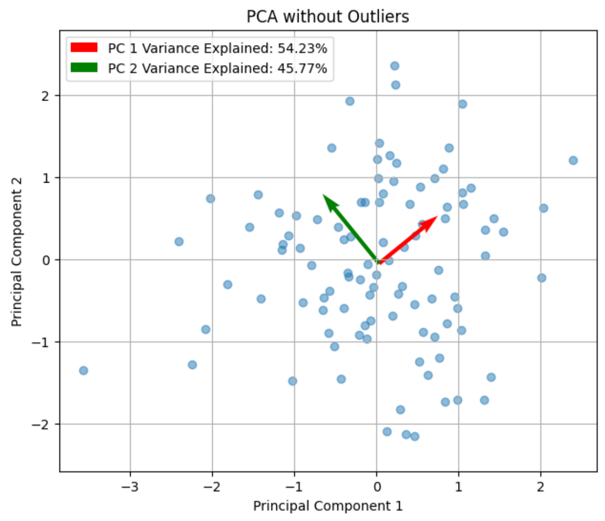
There are several ways to decide k, the number of principle components, from data that ensure an appropriate balance between capturing significant data structure and avoiding overfitting.

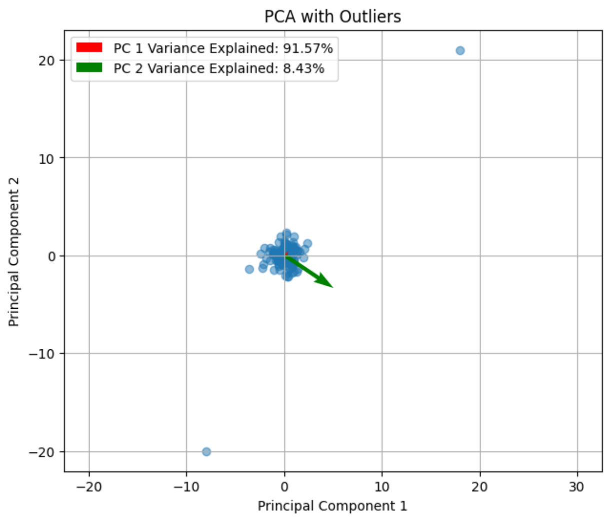
1. The total variance in the data can be calculated and PCA can be performed to determine the principal components. Then, a plot of the explained variance for each principal component can be made and a k can be chosen such that the cumulative variance explained by the first k components meets a desired threshold, typically between 85-95%. This is the same as calculating the explained variance ratio for each principal component which represents the proportion of the total variance in the data explained by each principal component.
2. A scree plot of the eigenvalues (variances) of the principal components in descending order can be made. We can find the optimal k by identify the “elbow point” where the rate of decrease in variance sharply slows down. The components before this point capture most of the variance in the data.
3. Another method is using cross-validation techniques where the data is split into training and validation sets. PCA can then be performed on the training set and both sets can be projected onto the first k principal components. The reconstruction error or classification performance can be evaluated as a function of k. We would choose the k that minimizes error or maximizes performance.

Question 1.5

Outliers can significantly impact the performance of PCA because PCA is sensitive to the variance in the data. PCA aims to find the directions that maximize the variance in the data. Outliers increase the overall variance and can cause PCA to align the principal components towards the outliers instead of the main data distribution. This misalignment can reduce the effectiveness of PCA in capturing the true structure of the data which leads to suboptimal dimensionality reduction and reconstruction.

An easy way to see this is through an example of PCA on randomly generated data.





As seen from the plots above, the presence of outliers has a significant impact on the PCA results. In the first plot without outliers, the explained variance ratio is much more balanced, with the first two principal components explaining roughly equal amounts of the variance in the data. However, in the plot with the outliers, the first principal component is heavily influenced by the outliers and the explained variance ratio is skewed. Here, the first principal component explains 91.57% of the variance, while the second component only explains 8.43% of the variance. This shows how outliers can significantly distort the PCA results.

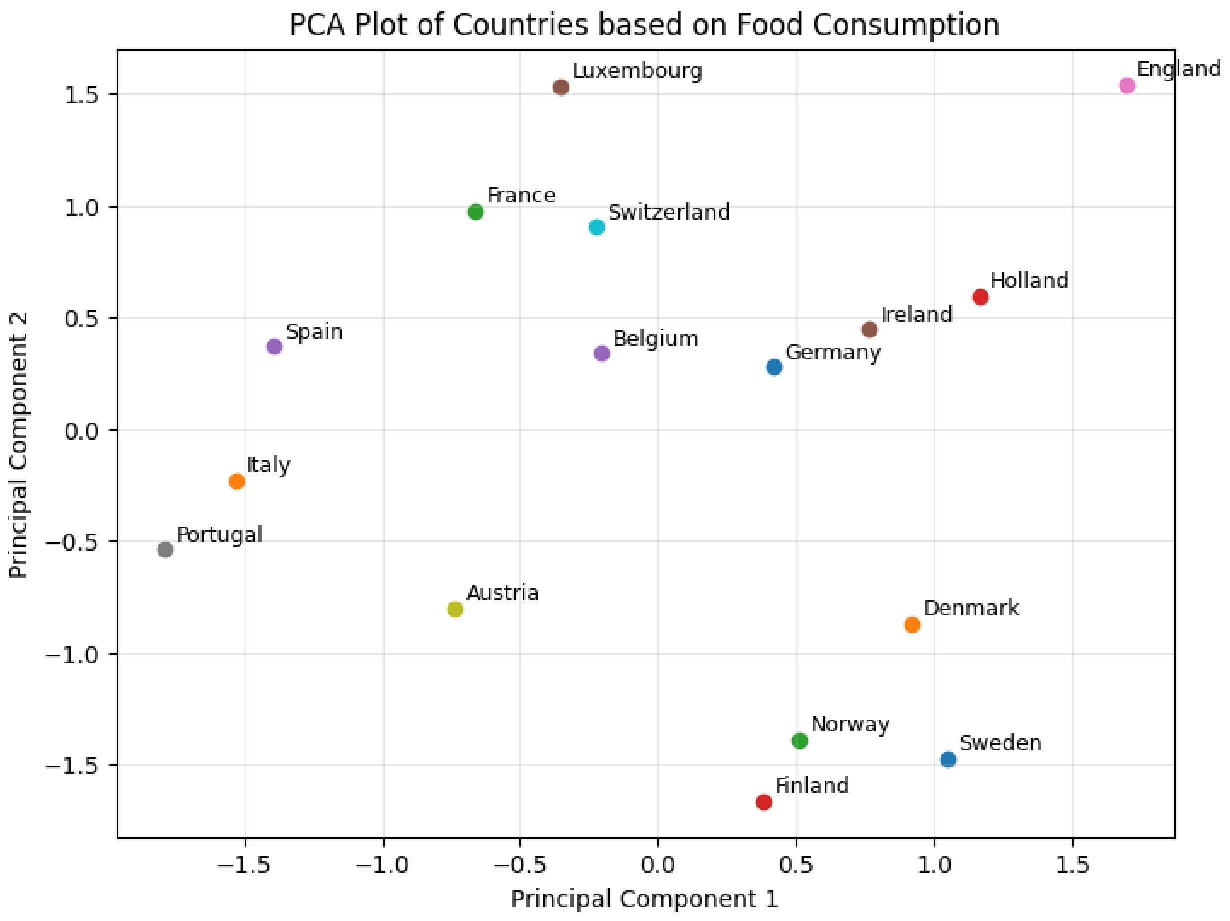
2 - PCA: Food consumption in European countries

The data `food-consumption.csv` contains 16 countries in Europe and their consumption for 20 food items, such as tea, jam, coffee, yogurt, and others. I will perform principal component analysis to explore the data. In this question, PCA will be implemented from scratch. The full code can be found in the zip file attached with this assignment.

2.A

First, PCA analysis will be performed on the data by treating each country's food consumption as their "feature" vectors. The goal is to find the weight vectors to combine the 20 food-item consumptions for each country. The data matrix will be set up in a way that each row represents a country and each column represents a type of food. For each row, or each country, there will be a value corresponding to the food consumption for each type of food.

The PCA code was developed using demo code from the lectures and can be found in the attached zip file.

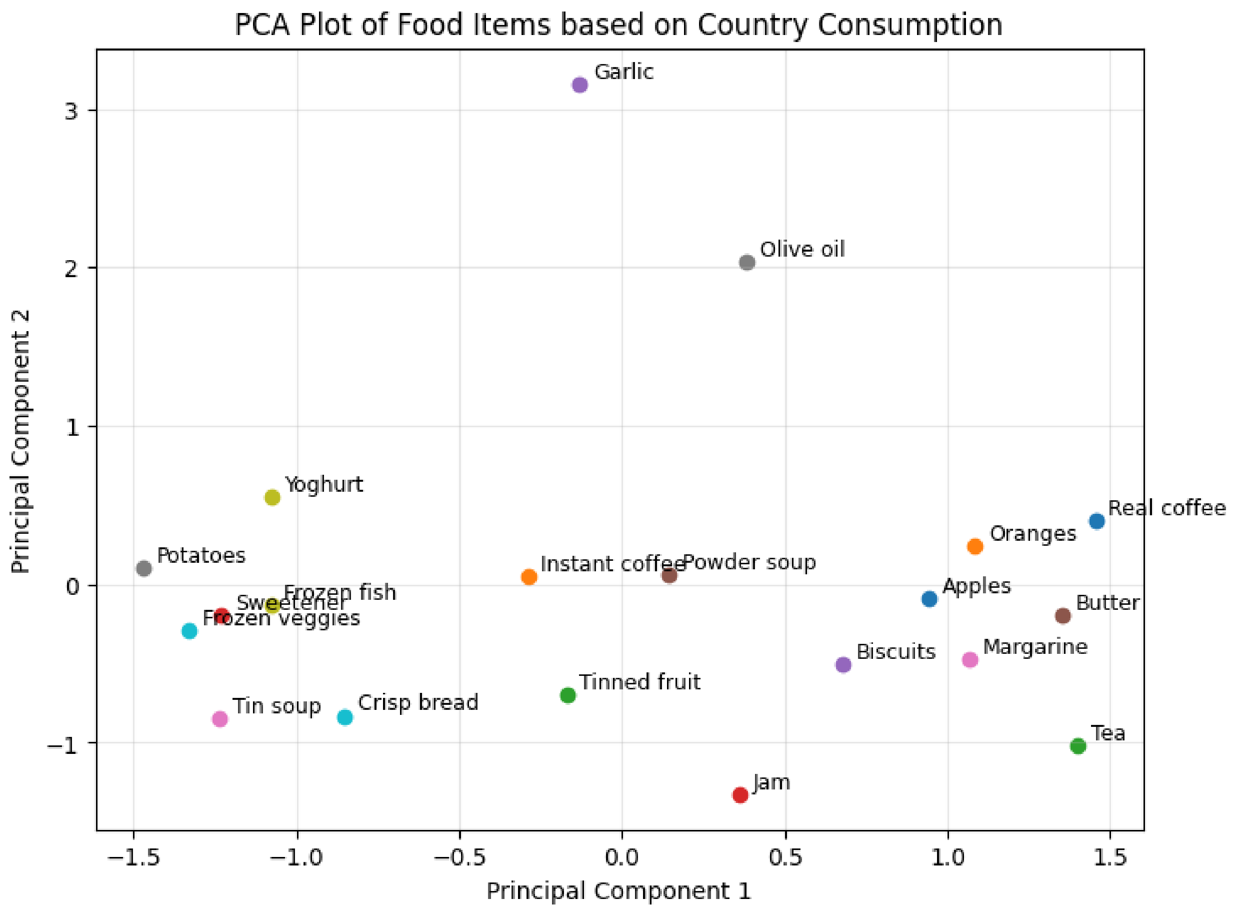


There are several interesting patterns when observing the plots above. There are many clusters that have formed where some countries are grouped together. Some countries that are geographically closer together are clustered together on the scatter plot, like Denmark, Norway, Sweden, and Finland, otherwise known as Scandinavia or the Nordic Countries. In the context of food consumption by country, this makes sense since their geographical location and cultural overlap might mean that they have similar eating habits. Another clustered group is Spain, Italy, and Portugal, who are also close together on the map and are countries associated with the Mediterranean diet. There are some countries with positive values (e.g., Luxembourg, England) and other countries with negative values (e.g., Finland, Sweden, Norway) which suggests that there are opposing patterns of food consumption among the countries. England seems to be an outlier, indicating that their food consumption patterns are different than the rest of the European countries.

2.B

Now, PCA analysis will be performed on the data by treating country consumptions as “feature” vectors for each food item. In other words, the goal is to find the weight vectors to combine country consumptions for each food item to perform PCA another way.

The PCA code was reran with the feature vectors transposed and the data labels adjusted to show food items. The code can be found in the attached zip file.



The PCA plot above shows a clear clustering and distribution of food items along the principal components. For example, fruits are grouped together (e.g. apples, oranges, margarine) far away from some frozen foods (e.g. frozen fish, frozen veggies) on the first principal component. Some food items having positive values (e.g., garlic, olive oil) and others have negative values (e.g., tea, butter, jam), suggesting that there are opposing patterns of consumption for these food items across the countries. Some food items appear to be outliers, such as garlic, olive oil and jam, indicating they have very distinct consumption patterns compared to the other food items. This makes sense since these items cannot be consumed alone and often supplement other food items. There also seems to be a distinction between processed/packaged food items (e.g., tin soup, frozen veggies, yoghurt) and fresh/natural food items (e.g., apples, oranges, tea) along the first principal component. Overall, the PCA results and the natural clustering of the food items makes sense. If a buyer prefers frozen foods such as frozen vegetables, they are likely to also purchase frozen fish. If they prefer fresh alternatives instead, they would buy items like apples and oranges, which are categorized separately and positioned further away.

3 - Order of faces using ISOMAP

This question aims to reproduce the ISOMAP algorithm results in the original paper for ISOMAP, J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323 that we have also seen in the lecture as an exercise.

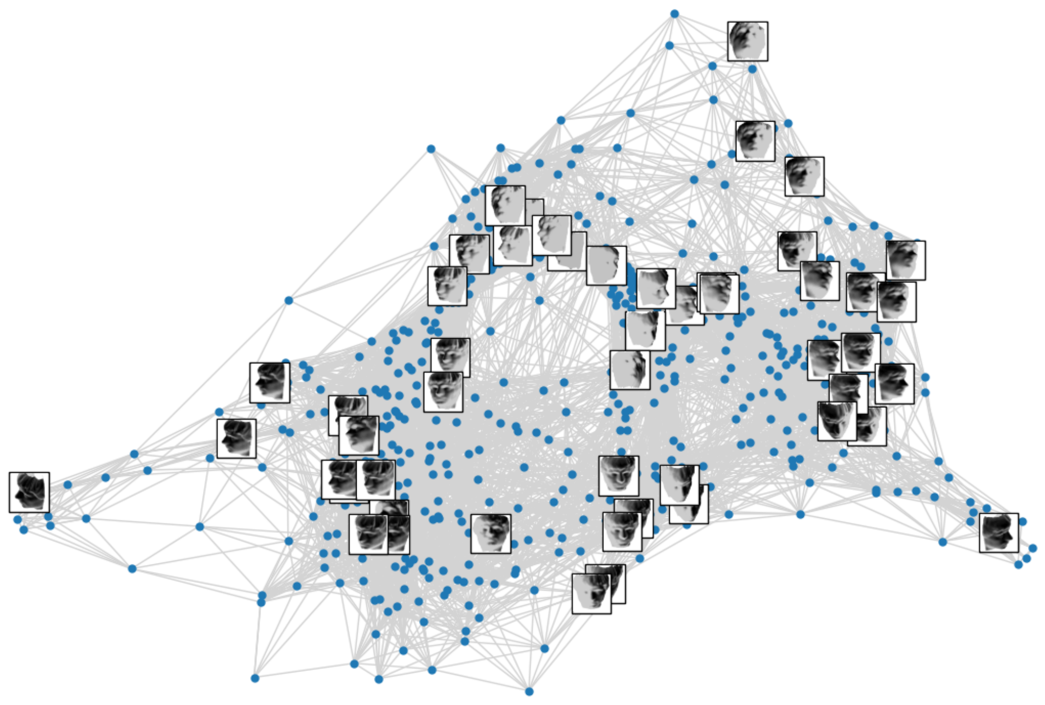
The file `isomap.mat` (or `isomap.dat`) contains 698 images, corresponding to different poses of the same face. Each image is given as a 64 x 64 luminosity map, hence represented as a vector in $\mathbb{R}^{4096}$. This vector is stored as a row in the file (this is one of the datasets used in the original paper).

By using Euclidean distance (in this case, a distance in $\mathbb{R}^{4096}$) to construct the ϵ-ISOMAP, I will tune the ϵ parameter to achieve the most reasonable performance. This is different from K-ISOMAP, where each node has exactly K nearest neighbors.

3.A

The `isomap.mat` data file was loaded in and an adjacency matrix was created. In order to create this adjacency matrix, I first had to code the ISOMAP algorithm from scratch and this code can be found in the attached zip file. The first step in ISOMAP is to build the weighted graph A which is the adjacency matrix. This matrix is created if the euclidean distance between points x^i and x^j was less than or equal to a tuned ϵ. After testing ϵ values from 1 to 50, I found that $\epsilon = 13$ worked the best in creating the Mickey Mouse shaped scatter plot. Therefore, this adjacency matrix was created with this fine tuned ϵ. I used the python library package, NetworkX, to visualize the nearest neighbor graph and plotted a few images corresponding to random data points.

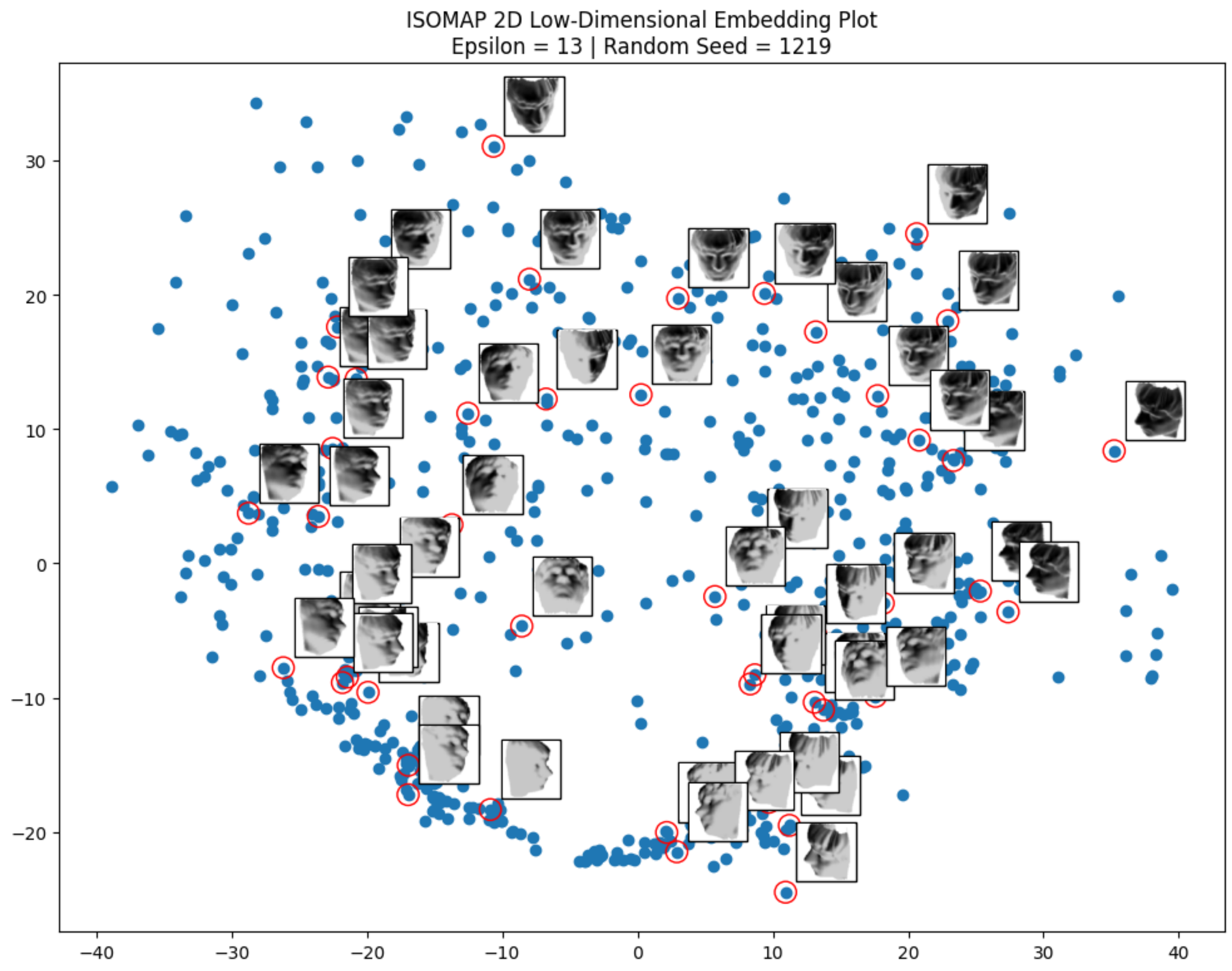
Nearest Neighbor Graph of the Adjacency Matrix using NetworkX
Epsilon = 13 | Random Seed = 645412



The plot above shows the nearest neighbor graph of the adjacency matrix using NetworkX. This is a visualization of the relationships between data points with the blue dots representing the data points and the gray lines connecting the nearest neighbors. There seems to be naturally clustered images sharing similar characteristics and some distinct groups or communities that are more densely connected within themselves. For example, images on the far left seem to be of faces looking left and are not lit as well as the images in the direct center of the network. Images on the far right are of faces looking right and are also dark due to low lighting. The images grouped in the center bottom seem to be of faces looking straight ahead with even lighting. Overall, we can see some groupings of images after simply just creating the adjacency matrix.

3.B

I ran the ISOMAP algorithm to obtain a two-dimensional low-dimensional embedding of the data and plotted the results on a scatter plot with a few images annotated near some random points.

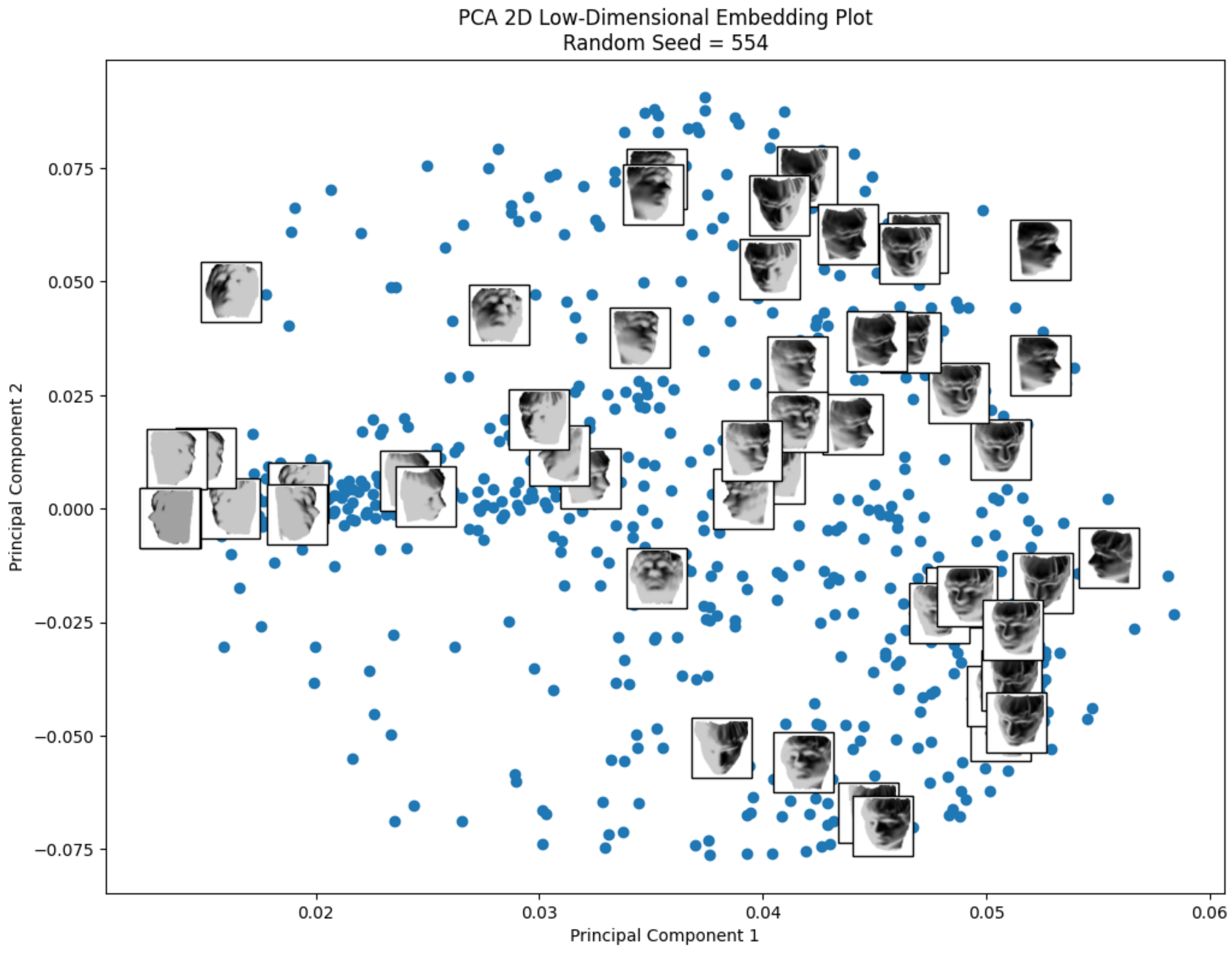


As seen from the plot above, the images are grouped in a similar way to the ISOMAP paper. Images on the top right are of faces looking towards the center of the plot (or faces looking bottom left), and seem to turn more left as you go around the plot in a clockwise direction. The images on the far right are of faces looking left while the images on the far left are of faces looking to the right. Images in the center are of faces looking straight ahead while images at the top seem to be of faces looking slightly down.

There seems to be grouping based on how the lighting is on the faces as well. Generally, well-lit faces are in the center while darker faces with more shadows and less lighting are on the outside edges of the plot. The data points are not as cleanly spread out as the ISOMAP paper and the algorithm does not group the faces looking up as well as the published paper did. Overall, the expected Mickey Mouse face can be visually seen from the data points and the ISOMAP algorithm clearly depicts separation of the images based on image characteristics.

3.C

Finally, I performed PCA on the images and projected them into the top 2 principal components. The scatter plot of the result is shown below.



The scatter plot shows similar groupings of images as the ISOMAP algorithm but it is visually harder to differentiate some changes. For instance, the bottom right section of the ISOMAP plot contains a cluster of faces looking down and to the left, while slightly above this cluster are faces looking down and to the right. The images on the far left are a mix of faces looking left and right, with the only similarity being that the faces are extremely lit and have many more white pixels than the faces on the far right who are not as well lit and have more shadows.

It seems that the PCA algorithm is grouping images based more on the brightness levels at different parts of the image while the ISOMAP algorithm captures not just the brightness levels but the compositional and directional aspects of the facial images. Overall, the ISOMAP algorithm provided a more meaningful projection compared to the PCA implementation.

4 - Density estimation: Psychological experiments

In *Kanai, R., Feilden, T., Firth, C. and Rees, G., 2011. Political orientations are correlated with brain structure in young adults. Current biology, 21(8), pp.677-680.*, data are collected to study whether or not the two brain regions are likely to be independent of each other and considering different types of political view. The data set `n90pol.csv` contains information on 90 university students who participated in a psychological experiment designed to look for relationships between the size of different regions of the brain and political views. The variables `amygdala` and `acc` indicate the volume of two particular brain regions known to be involved in emotions and decision-making, the amygdala and the anterior cingulate cortex; more exactly, these are residuals from the predicted volume, after adjusting for height, sex, and similar body-type variables. The variable `orientation` gives the students' locations on a five-point scale from 1 (very conservative) to 5 (very liberal). Note that in the dataset, we only have observations for orientation from 2 to 5.

Recall in this case, the kernel density estimator (KDE) for a density is given by

$$p(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{h} K\left(\frac{x^i - x}{h}\right),$$

where x^i are two-dimensional vectors, $h > 0$ is the kernel bandwidth, based on the criterion discussed in lecture. For one-dimensional KDE, I will use a one-dimensional Gaussian kernel

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

For two-dimensional KDE, I will use a two-dimensional Gaussian kernel: for

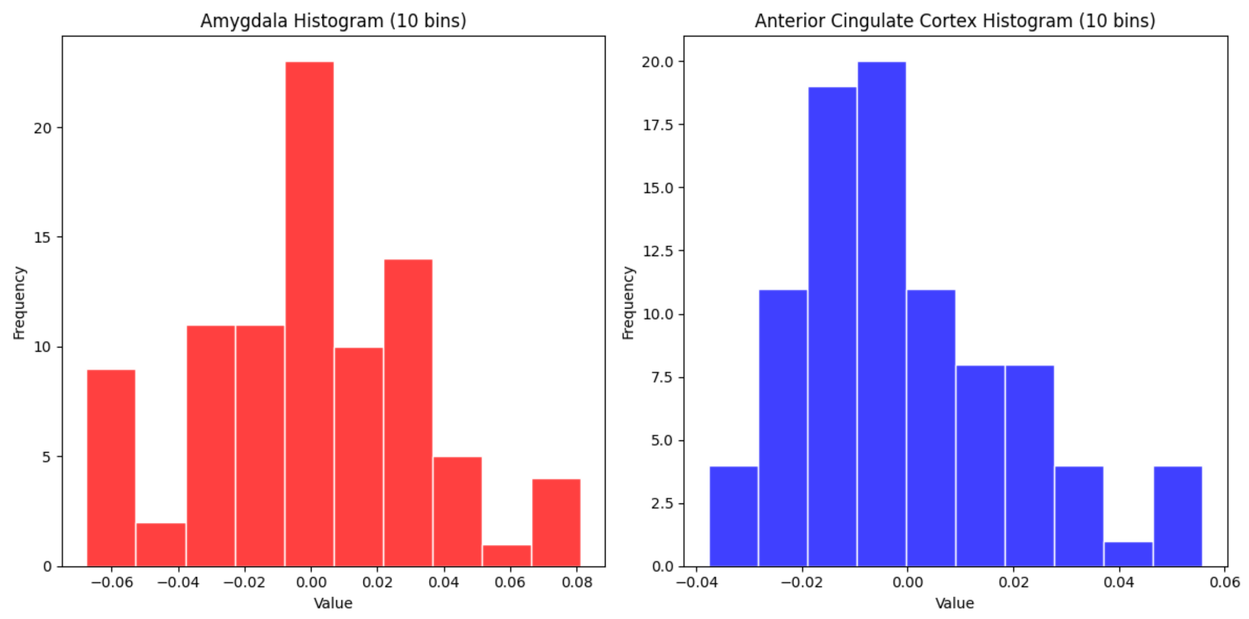
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2,$$

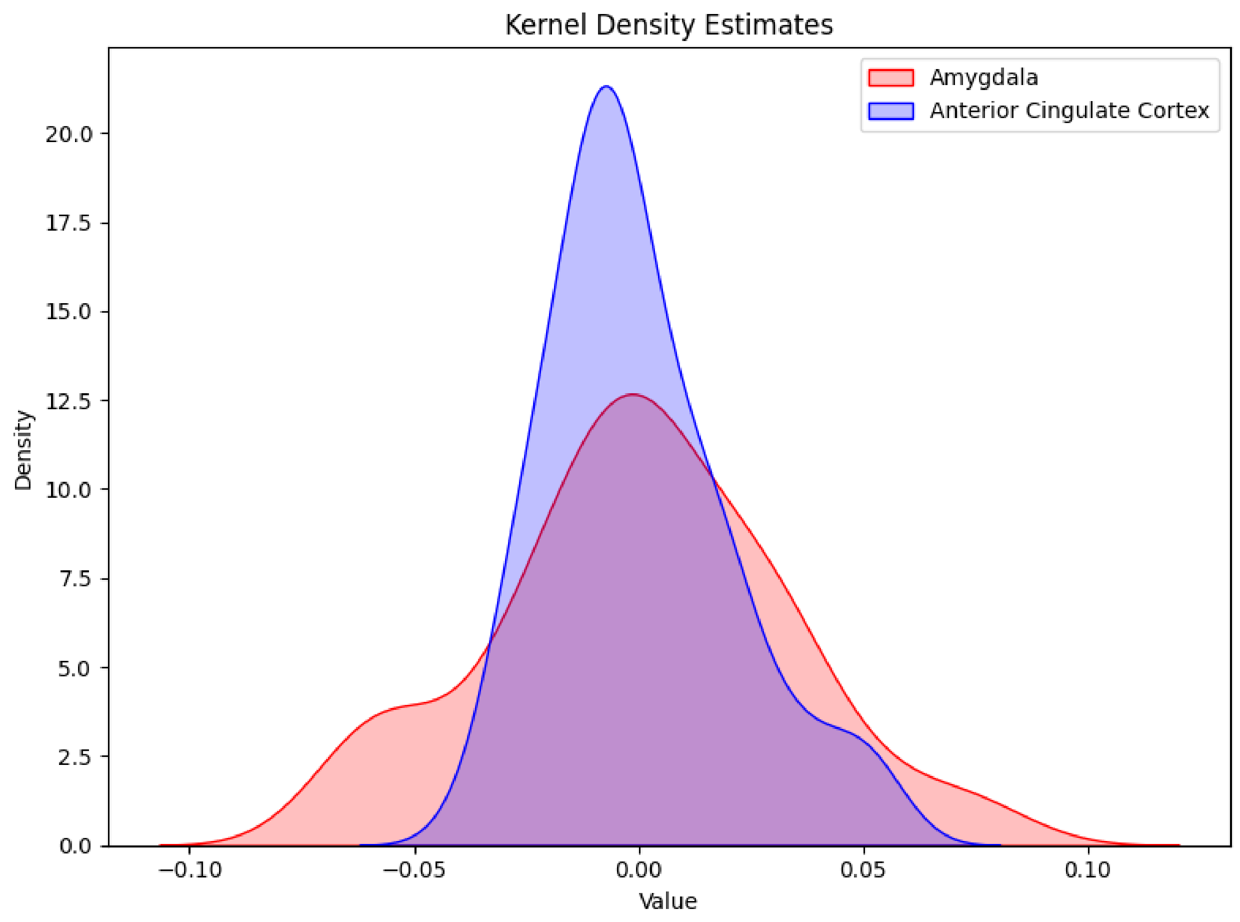
where x_1 and x_2 are the two dimensions respectively

$$K(x) = \frac{1}{2\pi} e^{-\frac{(x_1)^2 + (x_2)^2}{2}}.$$

4.A

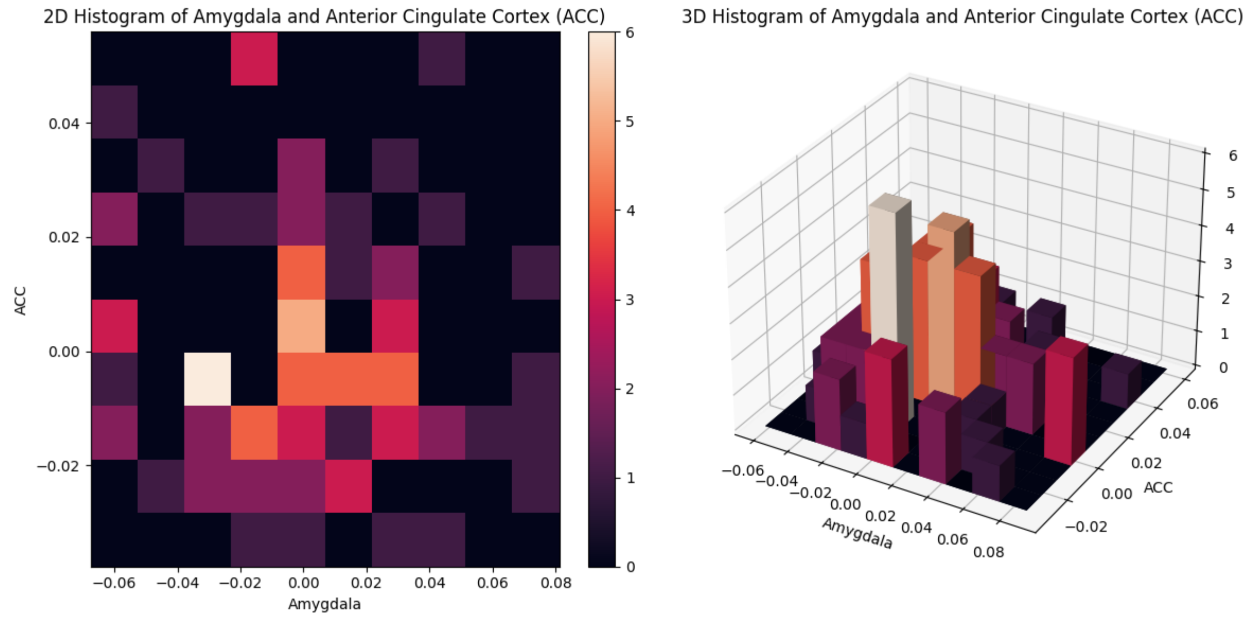
Question: Form the 1-dimensional histogram and KDE to estimate the distributions of `amygdala` and `acc`, respectively. For this question, you can ignore the variable `orientation`. Decide on a suitable number of bins so you can see the shape of the distribution clearly. Set an appropriate kernel bandwidth $h > 0$.


Amygdala Histogram (10 bins)
Frequency
Value
Anterior Cingulate Cortex Histogram (10 bins)
Frequency
Value


Kernel Density Estimates
🟥 Amygdala
🟦 Anterior Cingulate Cortex
Density
Value

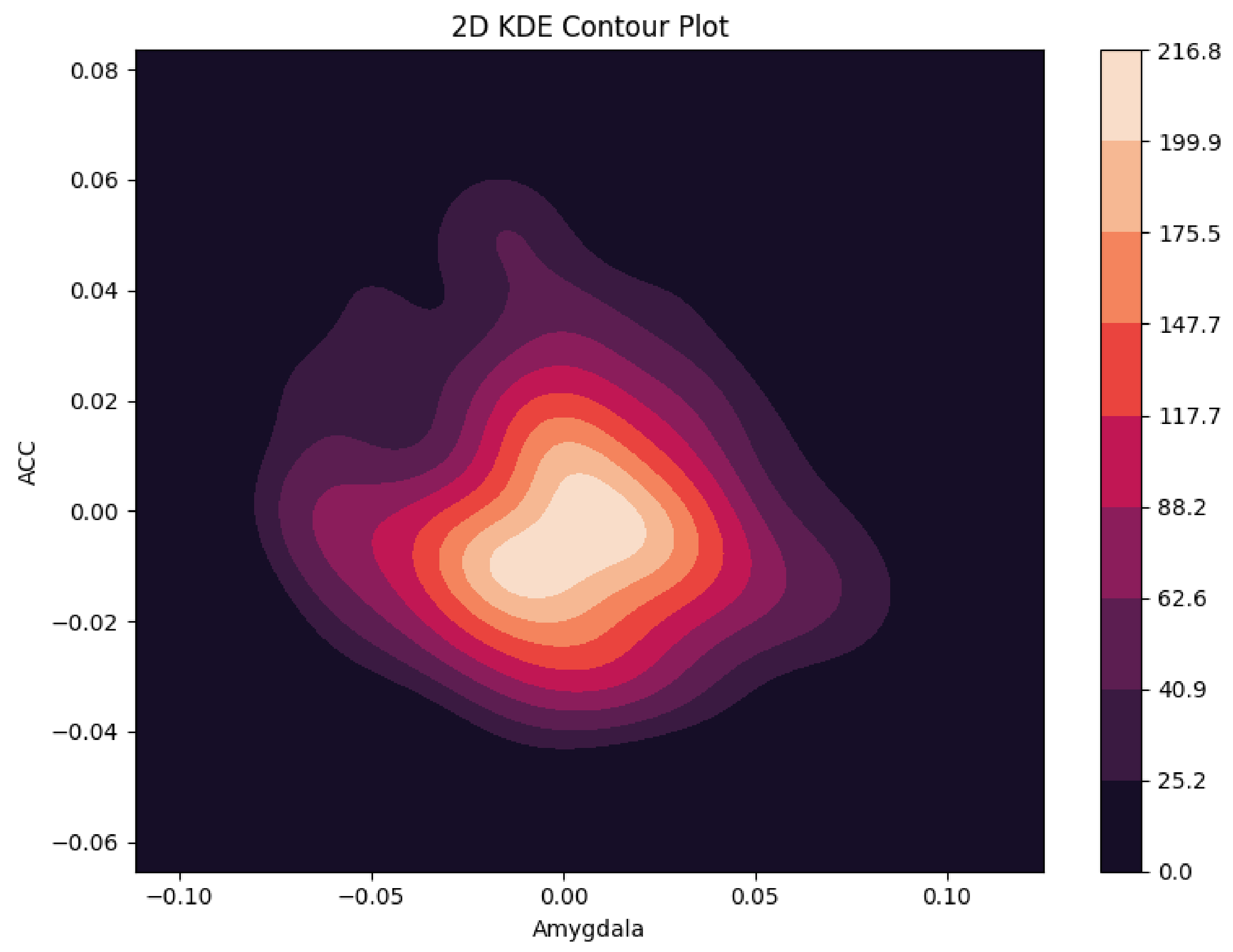
4.B

Question: Form 2-dimensional histogram for the pairs of variables (`amygdala`, `acc`). Decide on a suitable number of bins so you can see the shape of the distribution clearly



4.C

Question: Use kernel-density-estimation (KDE) to estimate the 2-dimensional density function of (`amygdala`, `acc`). Set an appropriate kernel bandwidth $h > 0$. Please show the two-dimensional KDE (e.g., two-dimensional heat-map, two-dimensional contour plot, etc.) Please explain what you have observed: is the distribution unimodal or bi-modal? Are there any outliers? Are the two variables (`amygdala`, `acc`) likely to be independent or not? Please support your argument with reasonable investigations.



As seen from the contour plot above, the data seems to be unimodal since there is only one peak in the distribution. The `bw_method` parameter in sns.kdeplot() is used to set the bandwidth, or the smoothing parameter, of the Gaussian kernel used in the 2D kernel density estimation above. At the current bandwidth of 0.45, there is one peak but a smaller bandwidth results in a more detailed, "peaky" density estimate which produces more peaks. Regardless, I believe that the data is unimodal after seeing the histograms and kde plots. The 2D KDE contour plot also does not indicate the presence of any obvious outliers. The distribution appears to be relatively smooth and continuous without any points or regions that stand out significantly from the overall density pattern.

To test whether the two variables (`amygdala`, `acc`) are independent or not, I carried out a pearson correlation test for independence and a paired sample t-test.

```
Pearson correlation coefficient: -0.128484
p-value: 0.2275

T-test Test statistic: 0.001556
p-value: 0.9988
```

The pearson correlation coefficient is close to 0 which indicates no linear relationship but the p-value is higher than the common significance level thresholds (e.g., 0.05, 0.01, or 0.001). Therefore, I can't definitively say that the variables are independent from this test. The other test was a t-test where the p-value is much greater than 0.05. This suggests that there is no statistically significant difference between the paired samples (amygdala and ACC). Since the p-value is very high (0.9315), I failed to reject the null hypothesis. Therefore, there is no significant difference between the amygdala and ACC measurements and they could be considered independent.

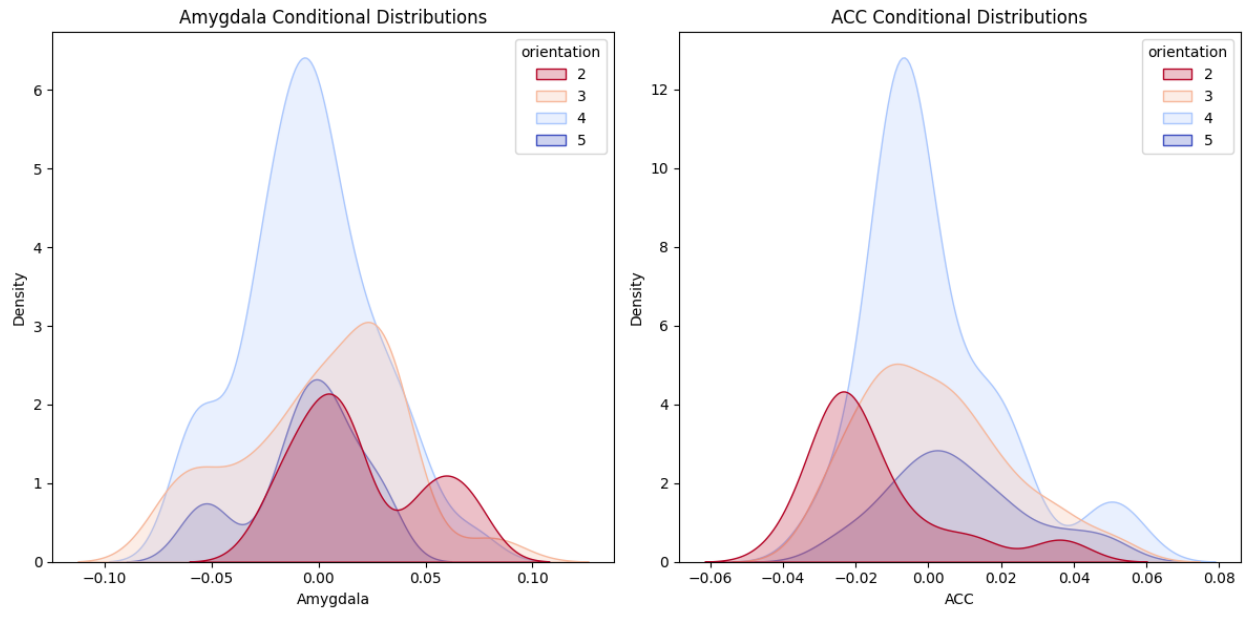
Regardless, the results indicate a very small correlation and the plots above suggest a degree of dependence between the two regions. This is expected as different parts of the brain often work together to process information and make decisions which would lead to some level of correlation.

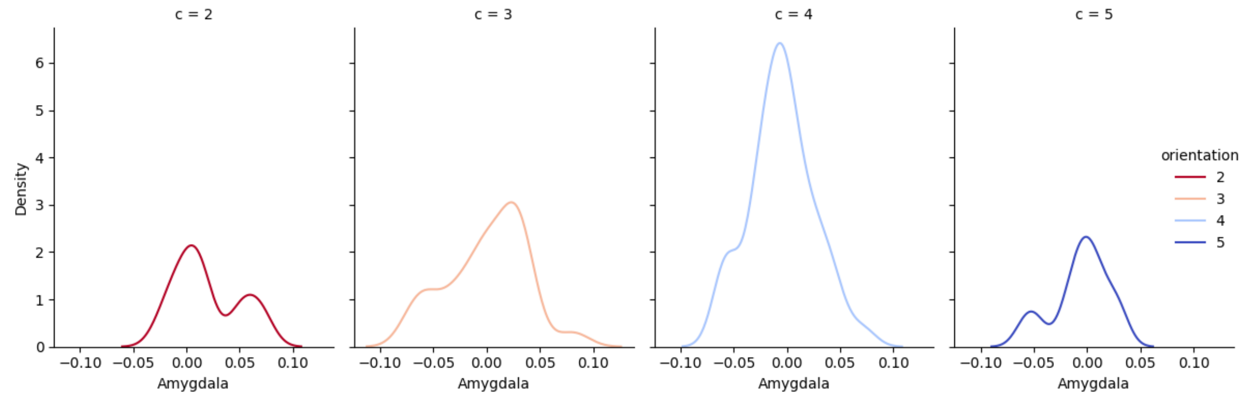
4.D

Question: We will consider the variable `orientation` and consider conditional distributions. Please plot the estimated conditional distribution of `amygdala` conditioning on political `orientation`: $p(\mathsf{amygdala}|\mathsf{orientation} = c)$, $c = 2, \dots, 5$, using KDE. Set an appropriate kernel bandwidth $h > 0$. Do the same for the volume of the `acc`: plot $p(\mathsf{acc}|\mathsf{orientation} = c)$, $c = 2, \dots, 5$ using KDE. (Note that the conditional distribution can be understood as fitting a distribution for the data with the same `orientation`. Thus you should plot 8 one-dimensional distribution functions in total for this question.)

Now please explain based on the results, can you infer that the conditional distribution of `amygdala` and `acc`, respectively, are different from $c = 2, \dots, 5$? This is a type of scientific question one could infer from the data: Whether or not there is a difference between brain structure and political view.

Please also fill out the *conditional sample mean* for the two variables.





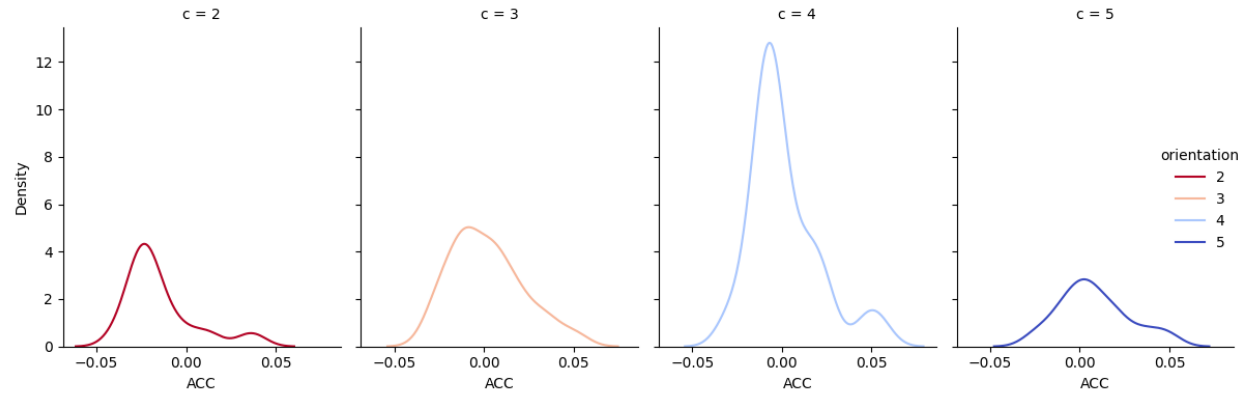


Table 1: Conditional Sample Means

Orientation	2	3	4	5
Amygdala	0.019062	0.000588	-0.004720	-0.005692
ACC	-0.014769	0.001671	0.001310	0.008142

Based on the results, I can infer that the conditional distribution of `amygdala` and `acc` are different for different political orientations. The amygdala distribution for extreme conservative views (c=2) is more spread out and bimodal, indicating greater variability. However, there isn't a clear indication that these individuals have consistently larger or smaller amygdala volumes. For more liberal views(c=4, c=5), the distributions are more centered around zero with c=4 having a notable peak. This suggests that individuals with liberal views might have amygdala volumes closer to the average. Similar to the amygdala, the ACC volumes for extreme conservative views (c=2) show a wider and slightly bimodal distribution, implying greater variability. As views become more liberal, the distributions again become more centralized around zero, particularly for c=4, suggesting ACC volumes closer to the average for these individuals.

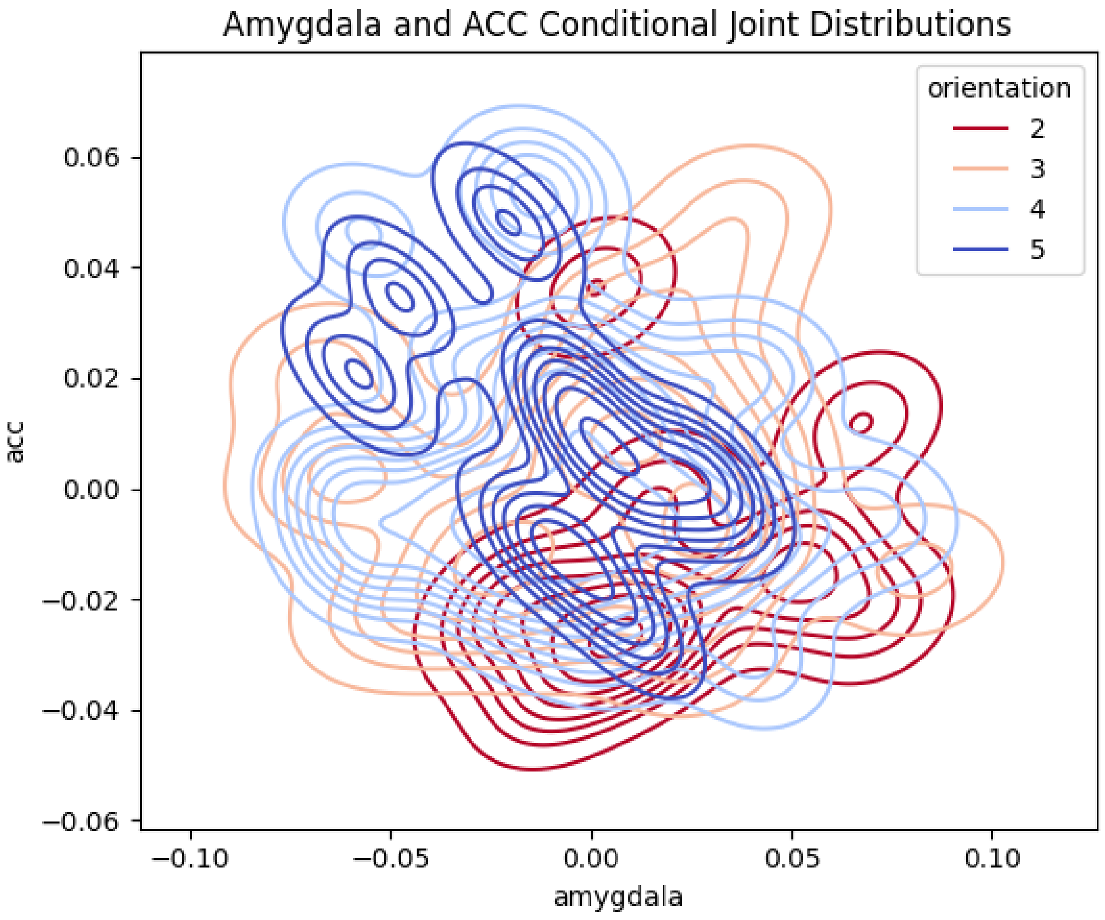
Orientation 3 and 4 were the most common which makes sense since most people are not on the extreme ends of the political spectrum. For data points with extreme views (c=2 and c=5), there is skewness in the data as evident by the table of the conditional sample means above. The amygdala values seem to positive for conservatives and negative for liberals, and the ACC values have a large difference between the conservative views (c=2) and the liberal views (c=5).

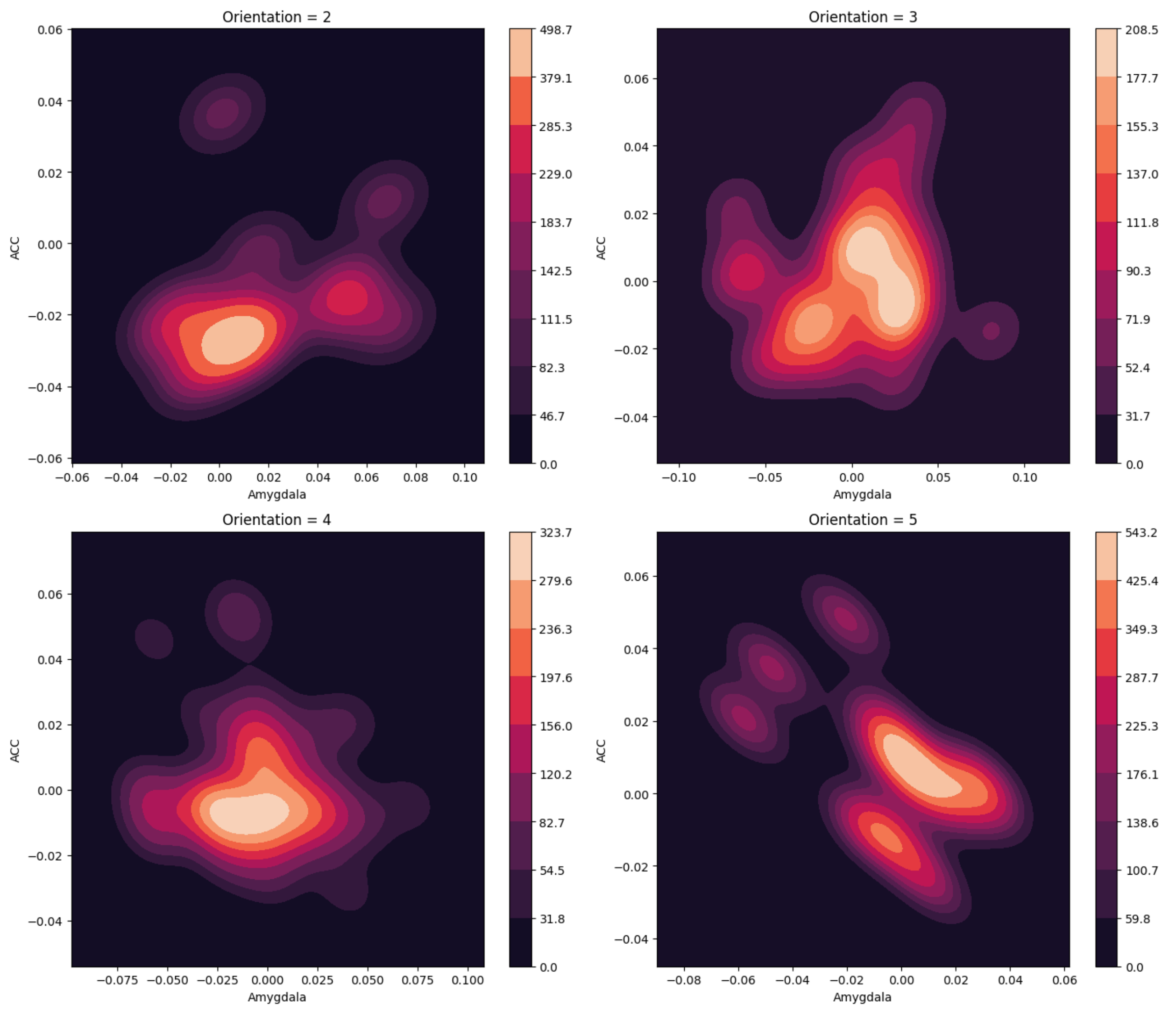
Overall, extreme conservative views are associated with a wider range of brain volumes, while liberal views are associated with more consistent, average volumes. However, the data does not clearly indicate that holding a specific political view is associated with having a definitively larger or smaller amygdala or ACC.

4.E

Question: Again we will consider the variable `orientation`. We will estimate the conditional *joint* distribution of the volume of the `amygdala` and `acc`, conditioning on a function of political `orientation`: $p(\mathsf{amygdala}, \mathsf{acc} | \mathsf{orientation} = c)$, $c = 2, \ldots, 5$. You will use two-dimensional KDE to achieve the goal; set an appropriate kernel bandwidth $h > 0$. Please show the two-dimensional KDE (e.g., two-dimensional heat-map, two-dimensional contour plot, etc.).

Please explain based on the results, can you infer that the conditional distribution of two variables (`amygdala`, `acc`) are different from $c = 2, \ldots, 5$? This is a type of scientific question one could infer from the data: Whether or not there is a difference between brain structure and political view.





Overall, I can infer that the conditional distributions of the two variables (`amygdala`, `acc`) are different for different political orientations. The contours for different orientations (c = 2, 3, 4, 5) above show distinct shapes and distributions, indicating that the joint distribution of amygdala and ACC volumes changes with political orientation. The plots show varying degrees of spread and clustering based on the political orientation. Conservatives (c = 2) show more variability while liberals (c = 5) display more concentrated and distinct clusters. The distributions for c = 3 and c = 4 are close, but the distributions for the extreme views (c = 2, c = 5) differ greatly in their shape and the regions they occupy on the plot.

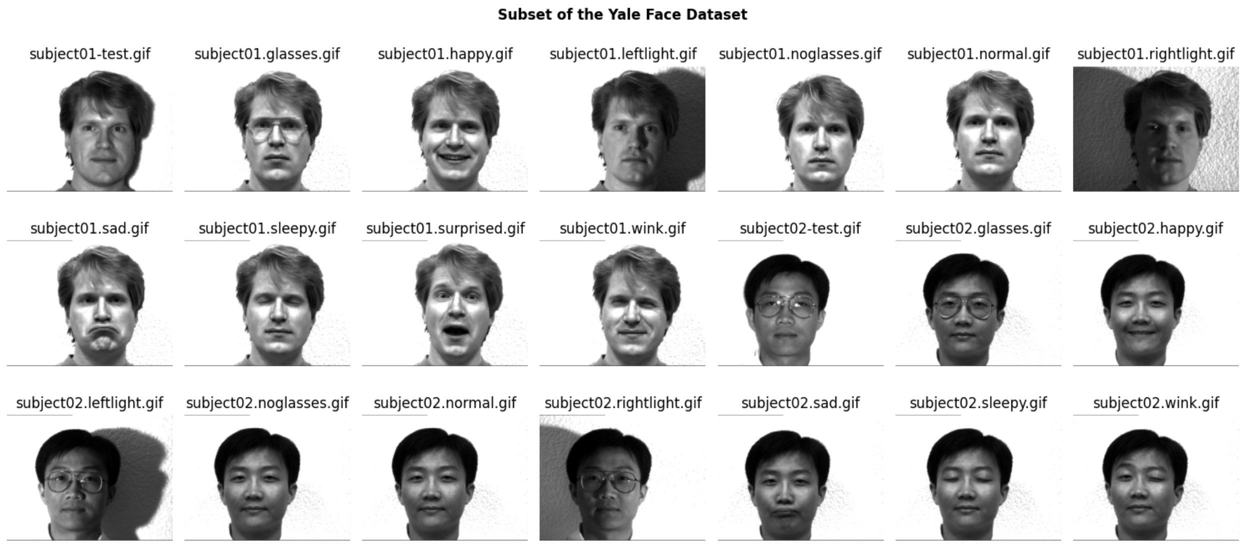
5 - Eigenfaces and simple face recognition

This part uses a subset of data from the Yale Face dataset and aims to use PCA for face recognition. The images are downsampled before being used in the PCA algorithm by a factor of 4 to turn them into a lower resolution image.

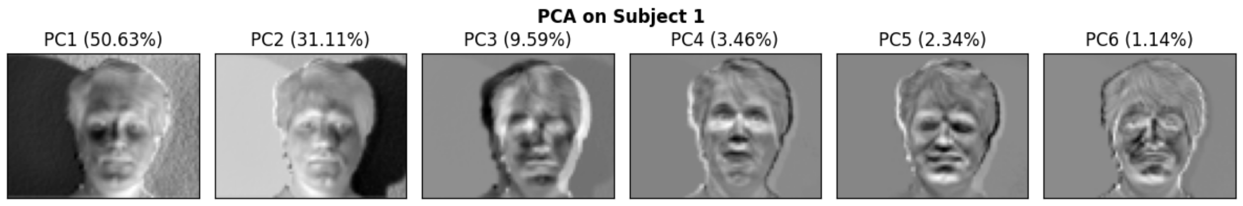
First, given a set of images for each person, we generate the eigenface using these images. I treated one picture from the same person as one data point for that person. Note that each image was first vectorized and was originally a matrix. Thus, the data matrix (for each person) is a matrix; each row is a vectorized picture. I found the weight vectors to combine the pictures to extract different "eigenfaces" that correspond to that person's pictures' first few principal components.

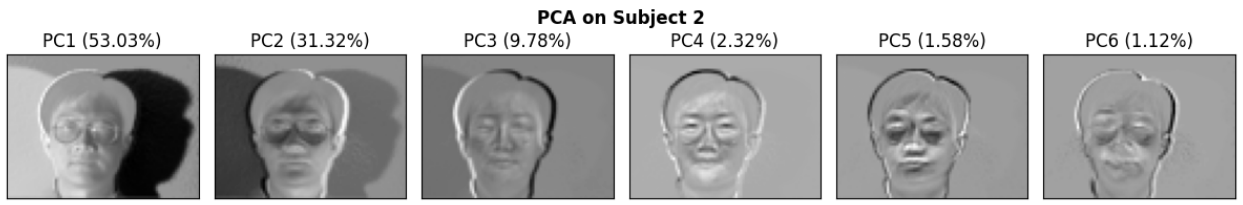
5.A

I performed PCA on the Yale face dataset for Subject 1 and Subject 2, respectively, using all the images except for two test images, one for each subject. All of the images in the subset of the dataset are shown below.



After running PCA, I reshaped the eigenvectors into proper images and the first 6 eigenfaces for each subject are shown below.





The first principal component (PC1), explaining around 50% of the variance, captures the overall structure of the face the best for both subjects. It captures prominent features like the eyes, nose, and mouth with clear distinction between the face which is the foreground and the background of a plain wall. PC2, accounting for around 30% of the variance, captures the variations in lighting or shading the best as well, particularly around the facial edges. One pattern that I noticed immediately was that the quality of the images showing each subject's face get worse for the latter principal components. Principal components like PC5 and PC6 explain very little variance, only accounting for around 1-2% of the total variance, and therefore resemble faces with features that are hard to make out. A lot of the clarity and facial features that separate humans are gone with each subsequent component, indicating that earlier components capture broader and more significant patterns.

5.B

For this part, I performed a simple face recognition task. Given the test image `subject01-test.gif` and `subject02-test.gif`, I first downsized by a factor of 4 (as before), and vectorized each image. I took the top eigenfaces of Subject 1 and Subject 2, respectively. Then, I calculated the *projection residual* of the 2 vectorized test images with the vectorized eigenfaces:

$$s_{ij} = \|(\text{test image})_j - (\text{eigenface}_i)(\text{eigenface})_i^T(\text{test image})_j\|_2^2$$

All four scores, s_{ij}, $i = 1, 2$, $j = 1, 2.$, are reported in the table below.

Table 2: Projection Residual Scores of Test Images with Eigenfaces

	Subject ID	Test Image ID	Projection Residual Score
s11	1	1	124,338.52
s21	2	1	272,193.01
s12	1	2	1,018,707.54
s22	2	2	861,580.77

Based on the results from the table above, we can see that test image 1 had a lower projection residual score when compared to subject 1's images (s11 < s21), indicating a better match with subject 1 than with subject 2. This is consistent with the prior knowledge that the test image 1 belonged to subject 1. Similarly, test image 2 showed a lower projection residual score against subject 2's images (s22 < s12), confirming that the face recognition algorithm correctly identified test image 2 as belonging to subject 2. This indicates that the algorithm can effectively and accurately classify new images for given subjects.

However, there needs to be a bigger difference in residual scores between test images and different subjects to enhance the algorithm's performance and increase classification confidence. One way to achieve this is by increasing the dataset with more images of each subject which would improve the training of the PCA algorithm. Additionally, rather than relying only on the top principal component, multiple top eigenfaces can be combined to capture more variance within the dataset. This would lead to better representation and differentiation of facial features which might improve classification accuracy.