Homework 2

Xiaofan Jiao

Question 1.1 & 1.2

Computation Data Analysis
HW2

1.1 1) Let X be an $m \times n$ matrix where each row represents a data point and each column represents a feature.
The sample covariance matrix S is defined as:

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

x^i is the i^{th} data point
μ is the mean vector

2) We aim to find the direction of w = max the variance of the data projected

3) Variance of the projections onto w is

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

4) Solve:

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

- subsitute S

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

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

- Thus

$$v = \arg\max_{\|w\|\le 1} w^TSw$$

- Thus $w^TSw = \lambda w^Tw = \lambda\|w\|^2$

5) We want to find the largest eigenvalue λ. Because $Sw = \lambda w$, we know that the weight vector we require for the first principal component direction must be the largest eigenvector. In this case, the eigenvector associated with the largest eigenvalue of the sample covariance S.

1.2 1) The probability density function of a Gaussian distribution is give

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

2) The likelihood function is the product of the individual densities:

$$L(\mu,\sigma^2) = \prod_{i=1}^{m} f(x^i;\mu,\sigma^2)$$

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

3) Taking the natural logarithm of the likelihood function

$$l(\mu, \sigma^2) = \log L(\mu, \sigma^2)$$

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

4) Simplify: $= \sum_{i=1}^{m} \left(\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \log\left(\exp\left(-\frac{(x^i-\mu)^2}{2\sigma^2}\right)\right)\right)$

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

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

5) To find the MLE for μ, we take the partial derivative & set $=0$

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

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

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

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

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

Thus $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^i$ the MLE for μ

6) To find the MLE for σ^2, same as step 5

$$\frac{\partial l}{\partial \sigma^2} = -\frac{m}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{m}(x^i-\mu)^2 = 0$$

$$-\frac{m}{\sigma^2} + \frac{1}{\sigma^4}\sum_{i=1}^{m}(x^i-\mu)^2 = 0$$

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

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

Question 1.3

Isometric Mapping is quite fascinating because it effectively captures the underlying structure of the data, which isn't always possible with traditional linear methods.

1. First, we start with neighborhood graph construction. The goal is to capture the local geometry of the data manifold. To do this, we determine a set of neighbors for each data point. There are two main ways to do this: using K-nearest neighbors (K-NN) or an ϵ-radius method. With K-NN, each point is connected to its K nearest neighbors. Alternatively, in the ϵ-radius approach, each point is connected to all points within a fixed radius ϵ.
2. Once we have our neighborhood graph, the next step is geodesic distance estimation. This step approximates the true manifold distances between points. We achieve this by computing the shortest path distances between all pairs of points on the constructed neighborhood graph. These shortest paths are usually found using algorithms such as Floyd-Warshall or Dijkstra's, which provide us with an accurate approximation of the geodesic distances on the manifold.
3. The final step is low-dimensional embedding via Multidimensional Scaling (MDS). Here, we try to find a low-dimensional representation of the data that preserves the estimated geodesic distances. We apply classical MDS on the matrix of geodesic distances to obtain the low-dimensional embedding. The output is a set of points in the lower-dimensional space that maintains the manifold's geometric structure as faithfully as possible.

Question 1.4

Method 1: The Explained Variance Method aims to capture the majority of the variance in the data. First, we compute the explained variance for each principal component. The explained variance indicates how much of the total variance in the data is captured by each principal component. Second, we plot the cumulative explained variance as a function of the number of principal components. Lastly, we choose k such that the cumulative explained variance reaches a satisfactory level, often 90% or 95%. This means selecting the smallest number of components that account for most of the variance in the data.

Method 2: The Scree Plot method provides a visual way to determine k. Plot the eigenvalues (which correspond to the amount of variance explained by each principal component) in descending order. Look for an "elbow" point in the plot, where the eigenvalues start to level off. The elbow point suggests a natural cut-off where adding more components contributes less to explaining variance. The number of components before this point is chosen as k.

Method 3: Cross-Validation evaluates how well various values of k perform in terms of generalization. Perform k-fold cross-validation on the dataset using different

numbers of principal components. Evaluate the model performance (e.g., classification accuracy, regression error) for each value of k. Choose k that provides the best performance or a balance between performance and model complexity.

Question 1.5

Given that PCA is extremely sensitive to anomalies in the data, outliers can have a substantial impact on the algorithm's performance.

```python
import numpy as np

# Dataset with and without outliers
X_no_outlier = np.array([
    [2, 3],
    [3, 4],
    [4, 5],
    [5, 6]
])

X_with_outlier = np.array([
    [2, 3],
    [3, 4],
    [4, 5],
    [5, 6],
    [100, 200]
])

# Function to compute covariance matrix
def compute_covariance_matrix(X):
    X_centered = X - np.mean(X, axis=0)
    covariance_matrix = np.cov(X_centered, rowvar=False)
    return covariance_matrix

# Compute covariance matrices
cov_no_outlier = compute_covariance_matrix(X_no_outlier)
cov_with_outlier = compute_covariance_matrix(X_with_outlier)

print("Covariance Matrix without Outlier:\n", cov_no_outlier)
print("Covariance Matrix with Outlier:\n", cov_with_outlier)
```

Without Outlier:

The eigenvalues and eigenvectors, which represent the primary components, are obtained by performing eigenvalue decomposition on this matrix. The likelihood of the eigenvalues being relatively near indicates that both components contribute to the variation in the data in a comparable way.

```
Covariance Matrix without Outlier:
 [[1.66666667 1.66666667]
 [1.66666667 1.66666667]]
```

With Outlier:

The outlier has a significant impact on this covariance matrix, causing the first principal component to align more closely with the outlier's direction. As a result, one

eigenvalue will grow noticeably larger than the other in this new covariance matrix, reflecting the disproportionate variance caused by the outlier.

```
Covariance Matrix with Outlier:
 [[1863.7 3774.4]
 [3774.4 7645.3]]
```

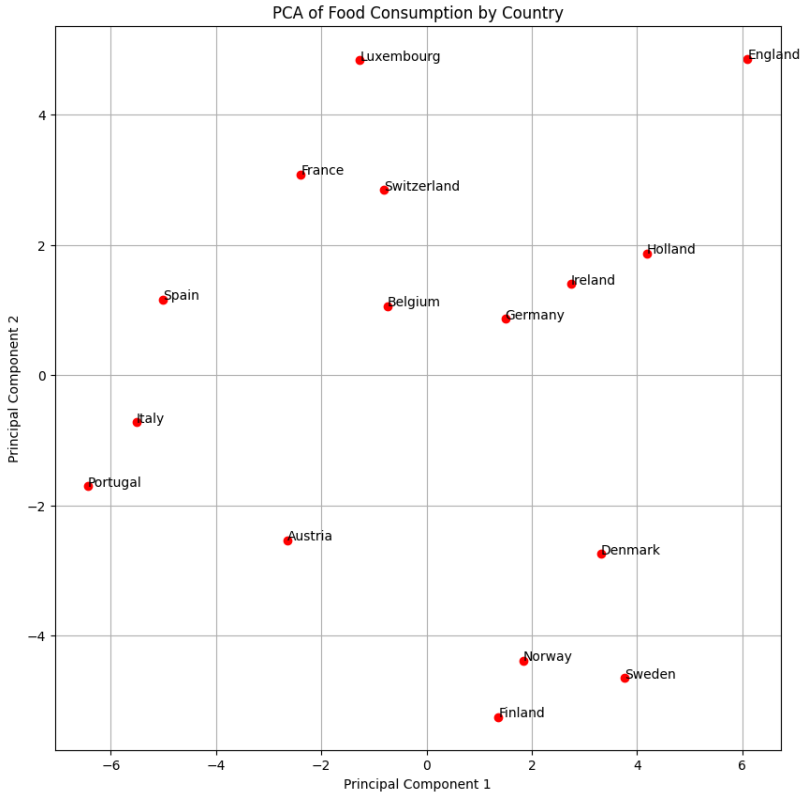
Outliers can have a substantial negative effect on the performance of PCA. They may lead to the first principal component explaining an excessively high amount of variation, falsely implying that the majority of the variance is impacted by the outlier in one manner or another. As a result, outliers can produce inaccurate findings when viewing or analyzing PCA-transformed data since the reduced dimensions might not adequately convey the relationships and structure of the data.

Question 2.1

The data is arranged in matrix form m×n, where m=16 corresponds to the countries, and n=20 corresponds to the different food items. Each country is represented by a row in the matrix, and each column represents a specific food item. If we consider food items as the features, we treat each row in the data matrix as a data point with 20 dimensions (different food items).

By performing PCA on this matrix, we reduce the dimensionality of the data from 20 to 2, extracting the two principal components that capture the most variance in the dataset. This allows us to represent each country's food consumption pattern as a two-dimensional vector.

From the plot, we observe several distinct patterns. For example, Holland and Portugal have nearly opposite food preferences. England stands out as the most isolated point, suggesting a unique food consumption style compared to the other countries. Countries like Belgium, Germany, and Ireland are clustered near the center, indicating similar food consumption habits.

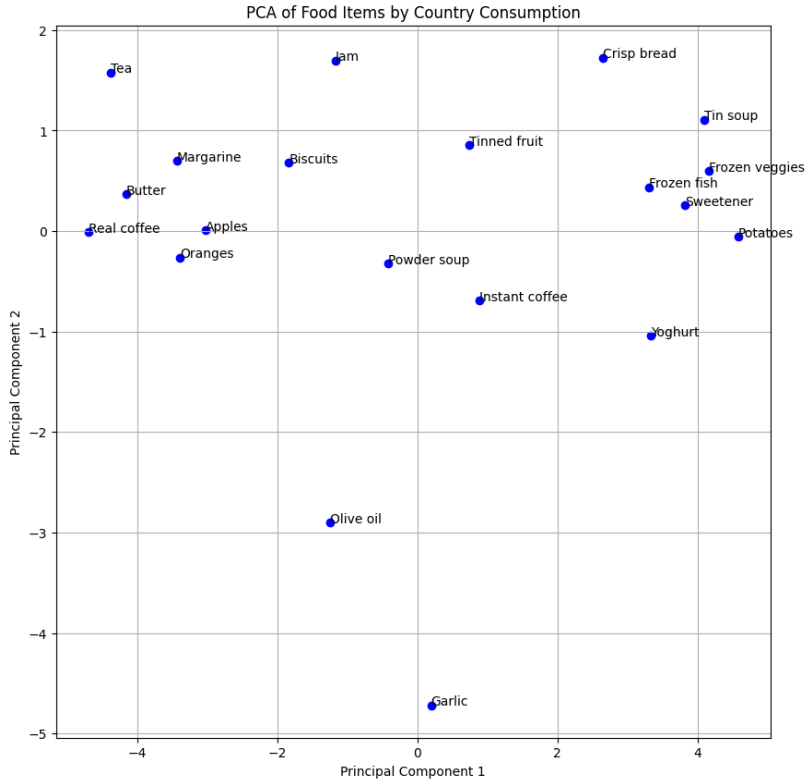


	PC1	PC2	Country
0	1.498626	0.869827	Germany
1	-5.512976	-0.720437	Italy
2	-2.386379	3.079790	France
3	4.185848	1.866264	Holland
4	-0.740965	1.061492	Belgium
5	-1.268932	4.830416	Luxembourg
6	6.102347	4.855348	England
7	-6.436008	-1.703052	Portugal
8	-2.647303	-2.531387	Austria
9	-0.812643	2.844445	Switzerland
10	3.764901	-4.645061	Sweden
11	3.312139	-2.742752	Denmark
12	1.841034	-4.380223	Norway
13	1.368684	-5.245961	Finland
14	-5.015538	1.162078	Spain
15	2.747167	1.399211	Ireland

Question 2.2

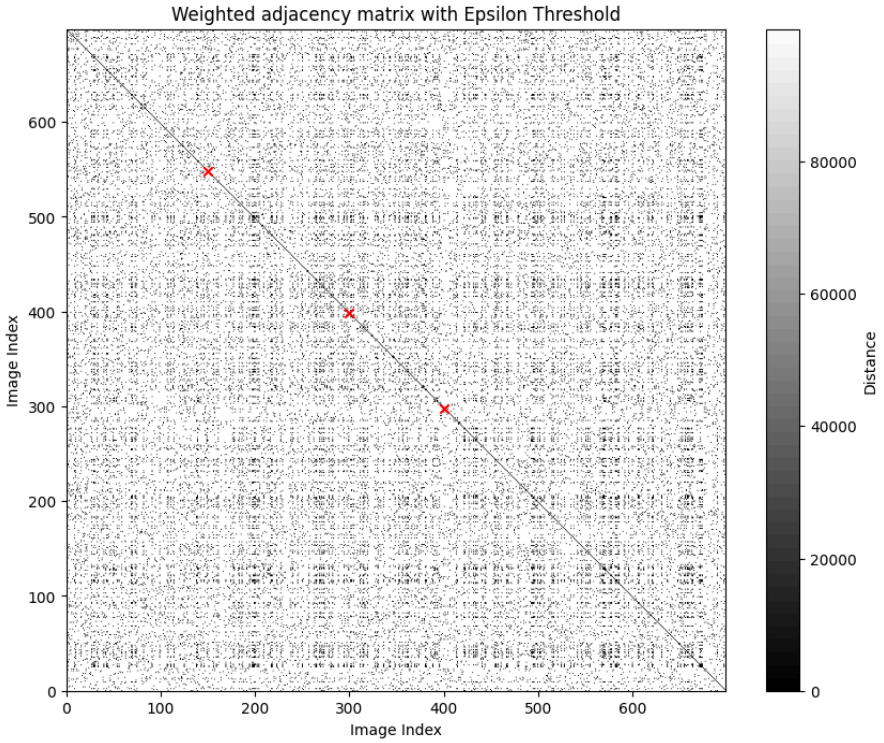
By performing PCA, we reduced the dimensionality from 16 to 2, extracting the two principal components that capture the most variance. The scatter plot shows the distribution of food items based on these components. In this plot, garlic and olive oil appear to have the most distinct preference patterns, where countries either strongly prefer or do not prefer these foods. Real coffee and tin soup have negatively correlated

preferences, indicating that countries tend to prefer real coffee and dislike tin soup, or vice versa. Additionally, some food items like tea, jam, and crisp bread form distinct clusters, indicating similar consumption patterns across countries, while yogurt is somewhat isolated, showing a unique consumption pattern.



	PC1	PC2	Food Item
0	-4.698147	-0.011183	Real coffee
1	0.878491	-0.689909	Instant coffee
2	-4.376306	1.577525	Tea
3	3.826781	0.254160	Sweetener
4	-1.835353	0.678379	Biscuits

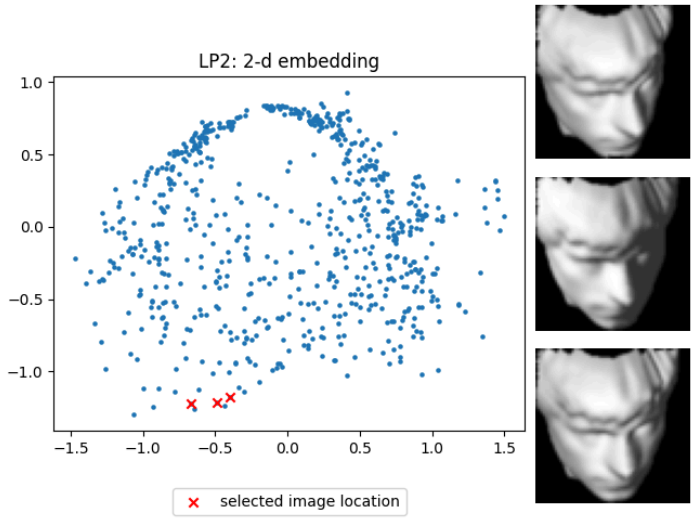
Question 3.1



Question 3.2

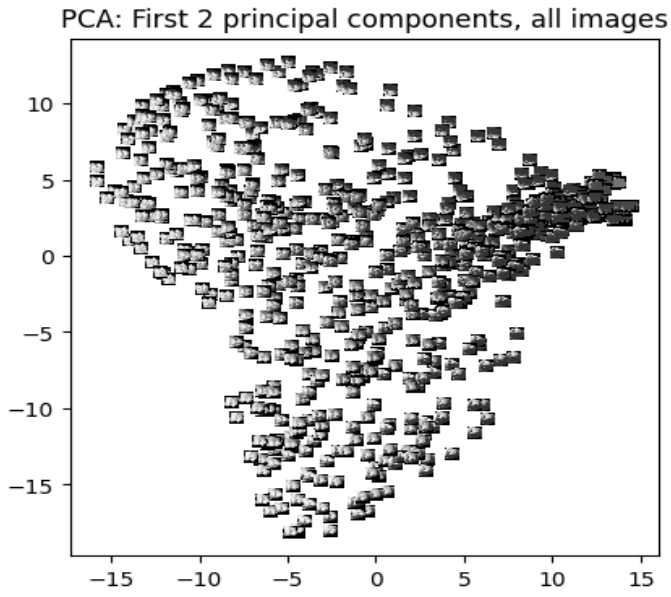
In the ISOMAP algorithm, the selected images (marked with red crosses) exhibit clear visual similarities, particularly in facial features, indicating that the algorithm effectively preserved local structures. The arrangement of these points in close proximity within the embedding space reflects the intrinsic geometry of the data, as similar high-dimensional images are clustered together. This outcome is consistent with the patterns observed in the relevant literature, where ISOMAP successfully maintains the global and local relationships among data points in the reduced dimensionality space.

The ISOMAP algorithm retained local structures as seen by the visual similarities across the selected photos (highlighted with red crosses), especially in facial features. When similar high-dimensional images are grouped together, it shows that the points are close to each other in the embedding space, highlighting the natural structure of the data. This result is in line with the trends shown in literature, where ISOMAP successfully preserves the local and global links between data points in the space of reduced dimensionality.



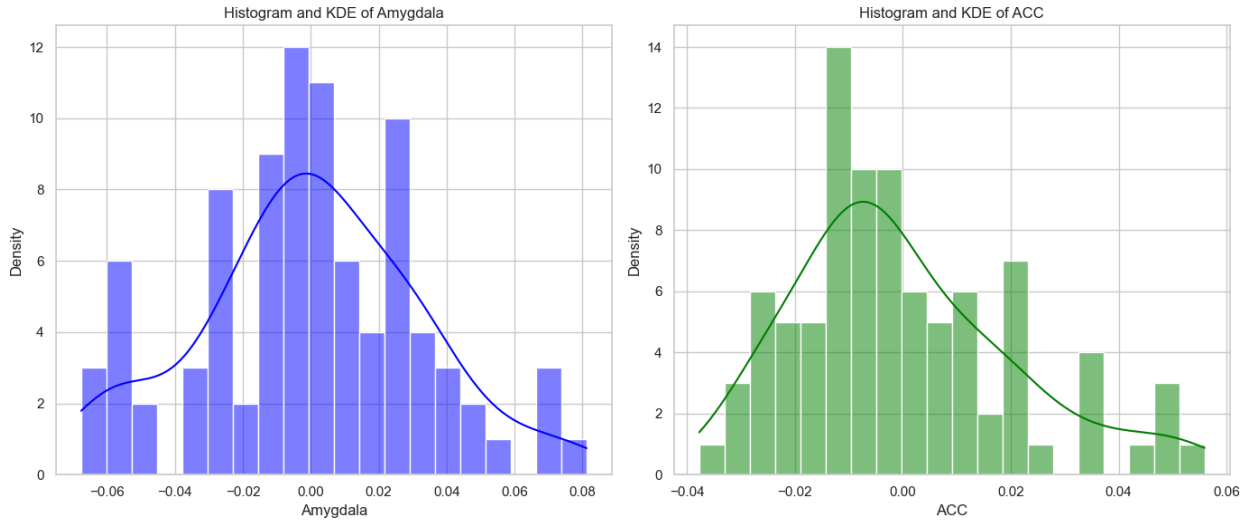
Question 3.3

When looking at the 2D projections of the images using ISOMAP and PCA, ISOMAP seems to give a more meaningful result by preserving the visual similarity and the underlying structure of the data. The ISOMAP scatter plot shows clusters of images that look similar, which means it effectively keeps the true distances and captures the natural shape of the data. On the other hand, the PCA scatter plot, although it reduces dimensionality by focusing on the top two principal components, shows a more spread-out arrangement with less clear grouping of similar images. This happens because PCA emphasizes maximizing variance along the principal components without considering the nonlinear relationships in the data, resulting in a less accurate representation compared to ISOMAP.

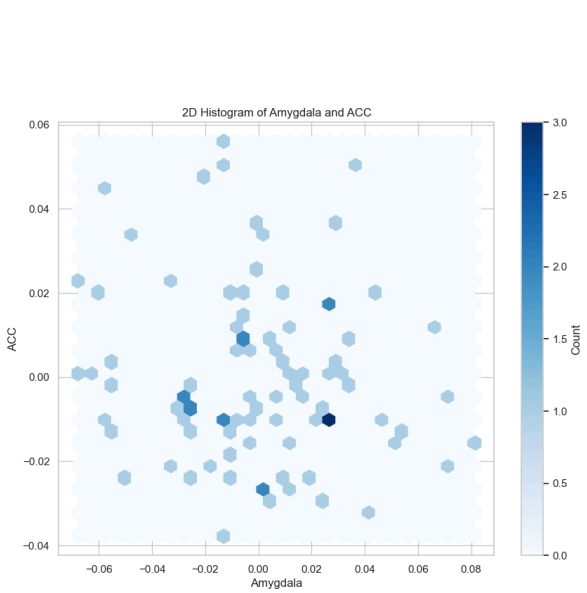


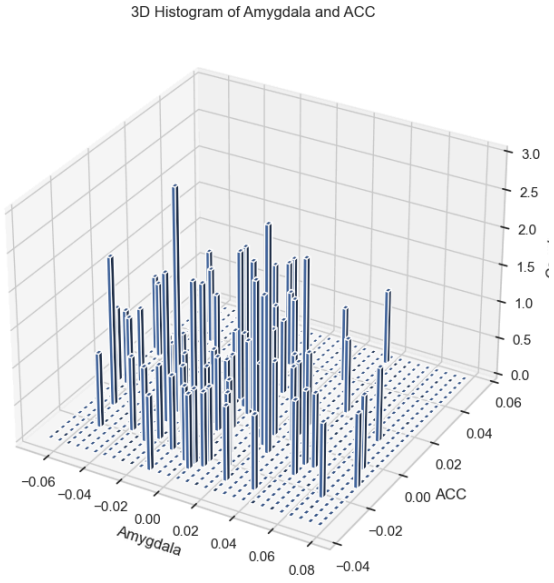
Question 4.1

The histograms display how often values occur, while the KDEs give a smoothed estimate of the data's probability density. For the KDE, I picked a kernel bandwidth h>0 that smooths the data just right. In the amygdala plot on the left, we see a slightly right-skewed distribution, while the ACC plot on the right shows a similar skewness but with a different density pattern.



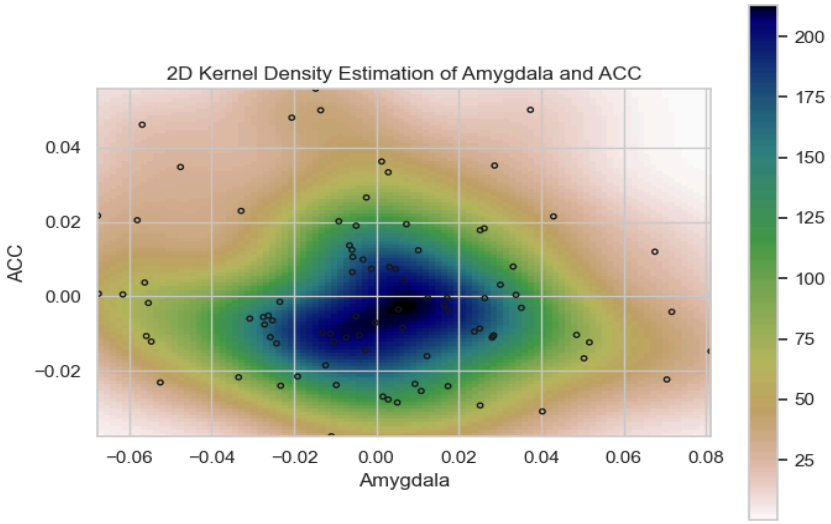
Question 4.2





Question 4.3

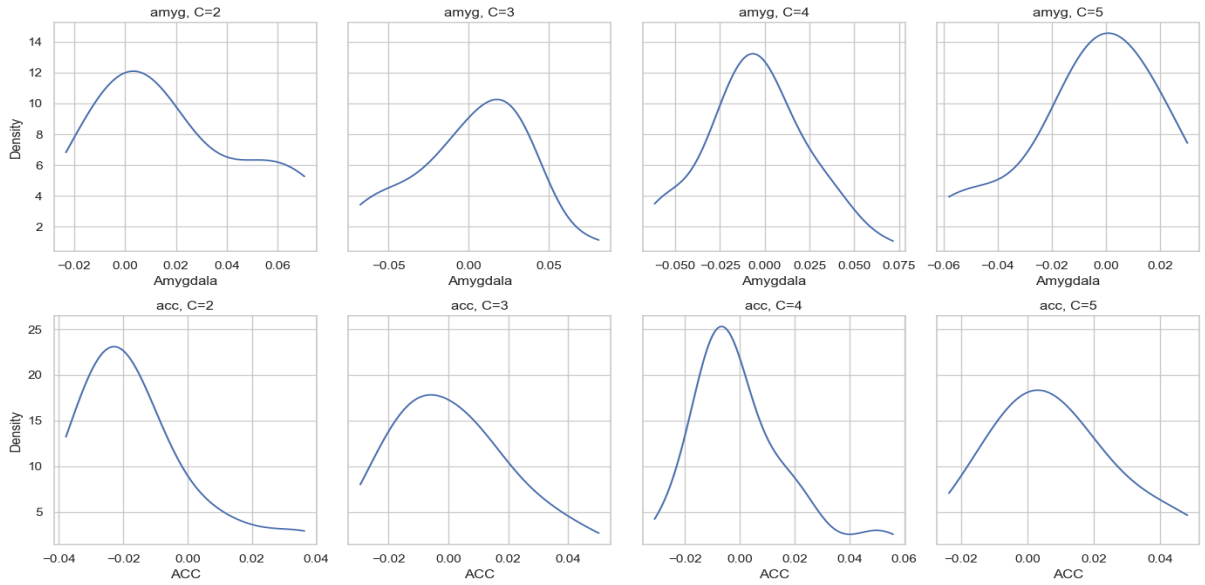
The 2-dimensional KDE heat map for the amygdala and ACC variables shows that the distribution has a single, dense central region, meaning it's primarily unimodal. This single peak indicates that most data points are clustered around the center, with density gradually decreasing as we move outward. We can see a few outliers around the edges, represented by individual points outside the main cluster. From the KDE plot, it seems that the amygdala and ACC variables are not independent because there's a clear central region with high density, indicating a relationship between the two variables.



Question 4.4

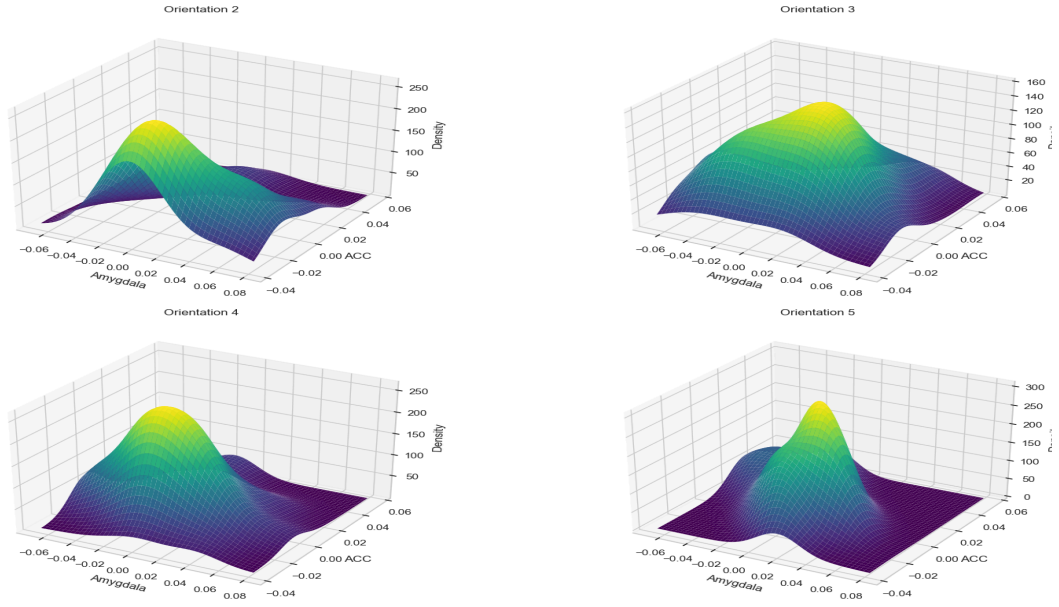
When looking at the conditional density plots, we see clear differences in the shapes of the distributions for each orientation. These patterns suggest that the distributions of both amygdala and ACC values are different depending on political orientation, hinting at a possible connection between brain structure and political views. The differences are further supported by the conditional sample means, which vary across different political orientations, showing that political orientation can influence brain structure. This means there are noticeable differences in brain structure associated with different political views.

	orientation	amygdala	acc
0	2	0.0191	-0.0148
1	3	0.0006	0.0017
2	4	-0.0047	0.0013
3	5	-0.0057	0.0081



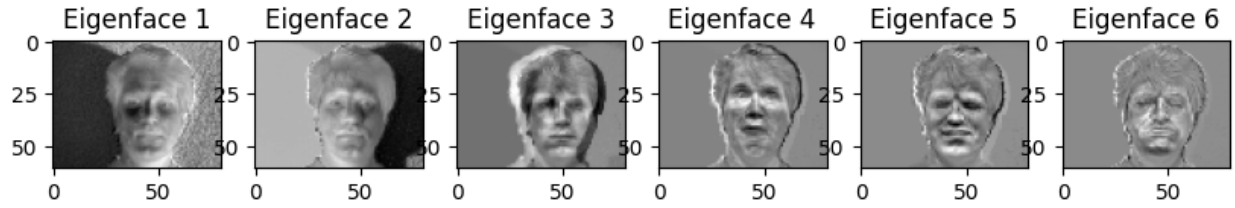
Question 4.5

When examining the 2-dimensional KDE plots for the joint distributions of amygdala and ACC, we notice clear differences in the density surfaces for each political orientation. Each plot shows how the amygdala and ACC values are distributed together for a specific political orientation. The shapes and peaks of these density surfaces change significantly, indicating that the joint distribution varies with political orientation. For instance, the locations and heights of the peak density values differ across the orientations. This suggests that the relationship between the amygdala and ACC is influenced by political orientation. These variations indicate that the distributions of the two variables are different depending on political orientation, hinting at a possible connection between brain structure and political views.

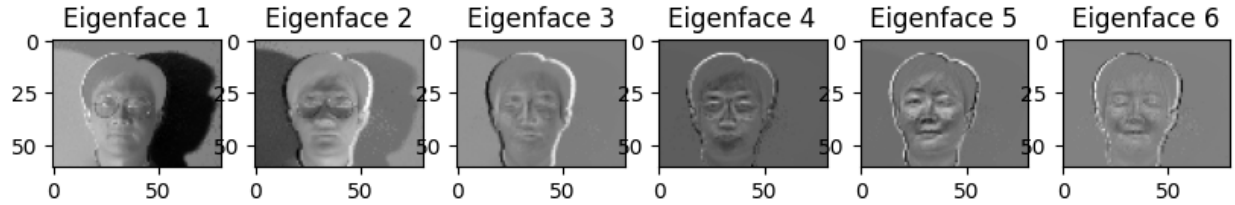


Question 5.1

Subject 1 shows distinct patterns, eigenface 1 captures the general shape and illumination of the face, while subsequent eigenfaces (2 to 6) capture more specific variations and details.



Similarly, for Subject 2, the eigenfaces also illustrate different aspects of facial variations. Eigenface 1 again captures the general facial structure and lighting. Eigenfaces 2 to 6 display more nuanced features, including variations in facial expression and specific facial features.



From the eigenfaces, we can observe that eigenfaces associated with higher eigenvalues are more identifiable. The patterns in the eigenfaces suggest that PCA effectively decomposes the facial images into orthogonal components that highlight the most significant features, enabling effective face recognition by focusing on these principal components.

Question 5.2

We calculated the projection residuals for the test images subject01-test.gif and subject02-test.gif using the top six eigenfaces for Subject 1 and Subject 2.

To recognize the faces of the test images using these scores, we compare the residuals: a smaller residual indicates a better match to the corresponding subject's eigenfaces. For subject01-test.gif, the smallest residuals occur for s_11, indicating it is most similar to Subject 1. For subject02-test.gif, the smallest residuals occur for s_22, indicating it is most similar to Subject 2.

The face recognition algorithm works reasonably well, as it correctly identifies the test images by comparing the residuals. However, improvements can be made by increasing the number of eigenfaces used, which may capture more variations and reduce the residuals. Additionally, incorporating more sophisticated preprocessing

techniques and exploring alternative dimensionality reduction methods could enhance the recognition accuracy.

Components	s_11	s_12	s_21	s_22
1	6742872.4473	32746524.6021	39491807.8305	3735545.1083
2	6183057.4747	32722164.2061	35096090.1528	3724889.2816
3	5843920.3683	32500816.1783	34591037.9533	2303889.4487
4	5813247.7261	32249128.0150	34272608.4077	2269497.1476
5	5739334.5250	31934665.1167	34089623.3663	2206158.5073
6	5679348.4437	30460607.7856	33791504.4035	1955858.0960