```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import fetch_lfw_people
faces = fetch_lfw_people(min_faces_per_person=60)
```

```
images = faces.images
data = faces.data
```

```
from sklearn.decomposition import PCA
```

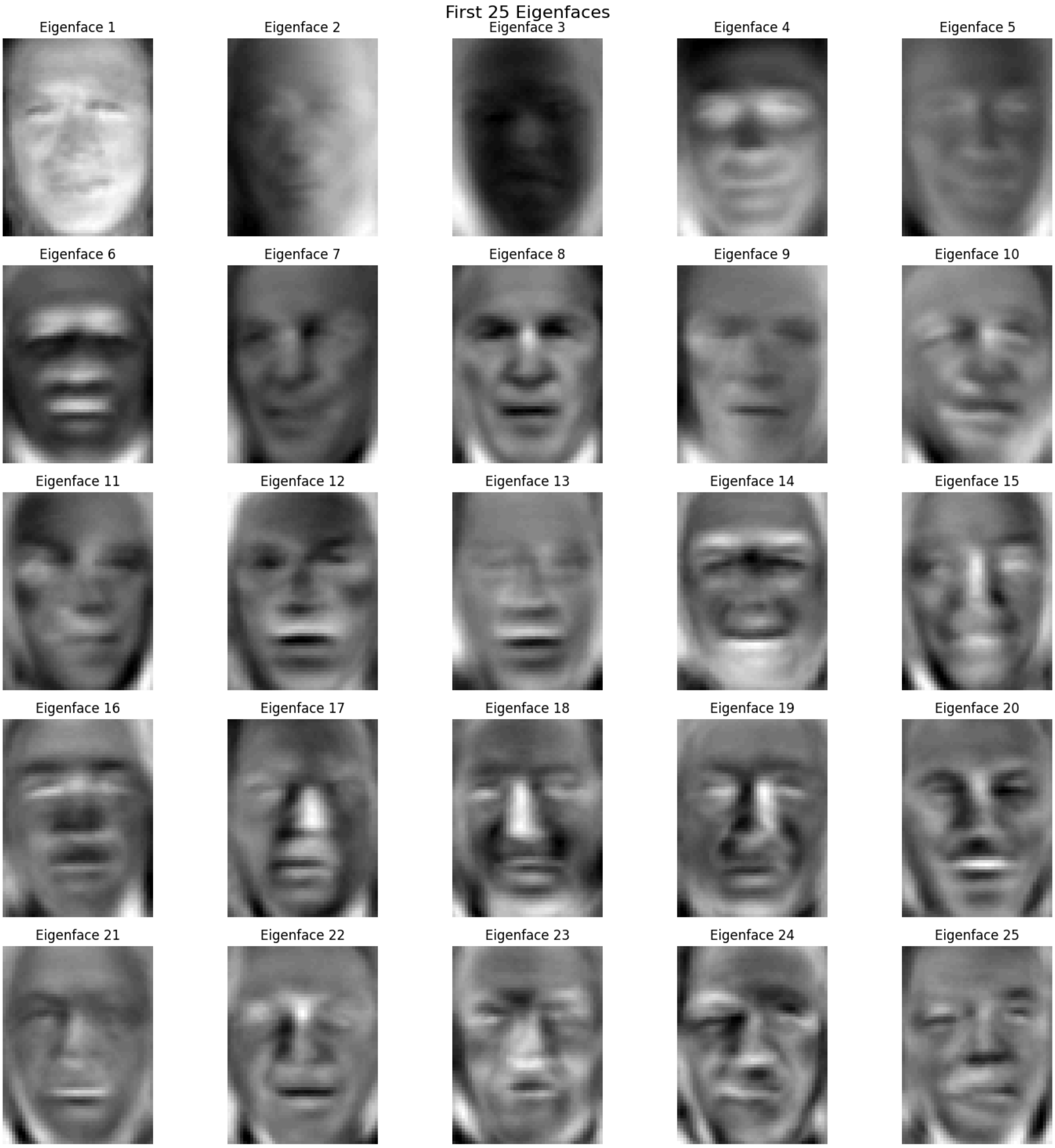
Q3)a Perform PCA on the dataset to find the first 150 components. Since this is a large dataset, you should use randomized PCA instead, which can also be found on sklearn. Show the eigenfaces associated with the first 1 through 25 principal components

```
pca = PCA(n_components=150, svd_solver='randomized')
pca.fit(data)
```

```
PCA
PCA(n_components=150, svd_solver='randomized')
```

```
eigen_faces = pca.components_[:25]
```

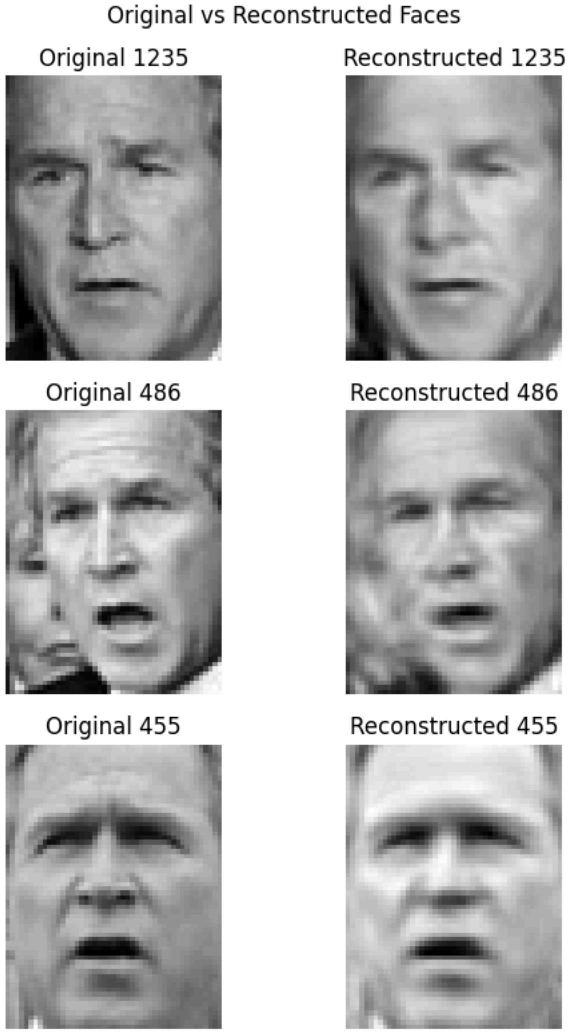
```
fig, axes = plt.subplots(5, 5, figsize=(15, 15))
fig.suptitle('First 25 Eigenfaces', fontsize=16)

for i, ax in enumerate(axes.ravel()):
    eigen_face = eigen_faces[i].reshape(faces.images[0].shape)
    ax.imshow(eigen_face, cmap='gray')
    ax.axis('off')
    ax.set_title(f'Eigenface {i+1}')

plt.tight_layout()
plt.show()
```



Q3)b Using the first 150 components you found, reconstruct a few faces of your choice and compare them with the original input images.

```
low_dim = pca.transform(data)
recon = pca.inverse_transform(low_dim)

n = 3
ind = np.random.choice(range(len(images)), n, replace=False)

fig, axes = plt.subplots(n, 2, figsize=(6, 8))
fig.suptitle('Original vs Reconstructed Faces', fontsize=12)
```

```
for i, idx in enumerate(ind):
  axes[i, 0].imshow(images[idx], cmap='gray')
  axes[i, 0].set_title(f'Original {idx}')
  axes[i, 0].axis('off')

  reconstructed_image = recon[idx].reshape(images[idx].shape)
  axes[i, 1].imshow(reconstructed_image, cmap='gray')
  axes[i, 1].set_title(f'Reconstructed {idx}')
  axes[i, 1].axis('off')

plt.tight_layout()
plt.show()
```



Hence we have performed PCA by reconstructing on 150 components and compared their transform with respect to original image