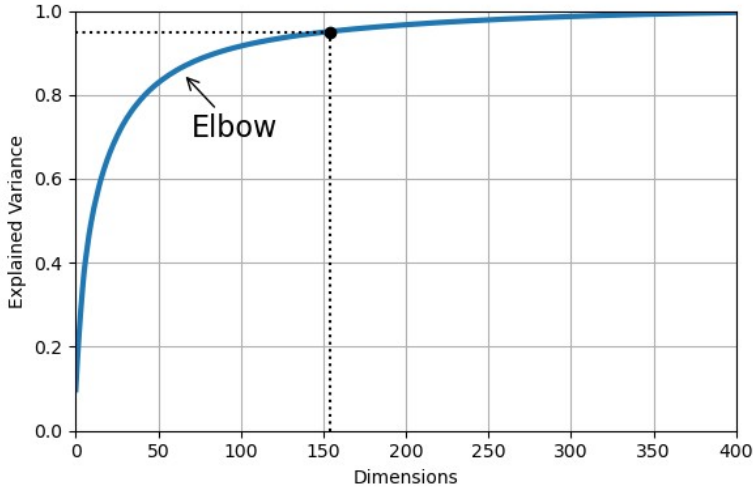
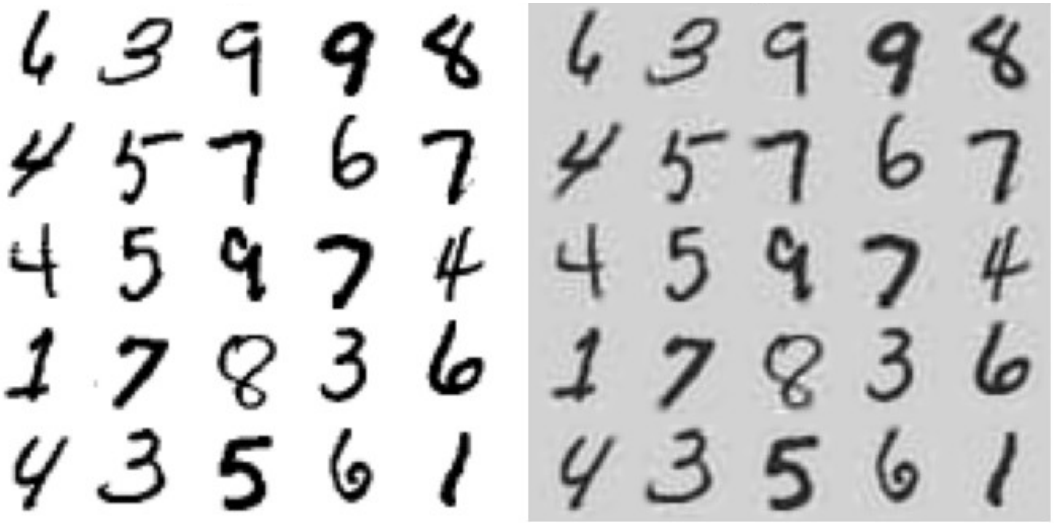
Q1-4. MNIST Compression with PCA and Incremental PCA

```
import numpy as np
import os

import matplotlib as mpl
import matplotlib.pyplot as plt

from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.decomposition import IncrementalPCA

# 그래프 결과를 저장한 경로 정의
PROJECT_ROOT_DIR = '.'
CHAPTER_ID = 'dim_reduction'
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, 'images', CHAPTER_ID)
os.makedirs(IMAGES_PATH, exist_ok=True)

# 그래프 결과를 저장하는 함수
def save_fig(fig_id, tight_layout=True, fig_extension='png', resolution=300):
    path = os.path.join(IMAGES_PATH, fig_id + '.' + fig_extension)
    print('Save Image ', fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)

# MNIST 데이터셋을 전시하는 함수
def plot_digits(instances, images_per_row=5, **options):
    size = 28
    images_per_row = min(len(instances), images_per_row)
    images = [instance.reshape(size,size) for instance in instances]
    n_rows = (len(instances) - 1) // images_per_row + 1
    row_images = []
    n_empty = n_rows * images_per_row - len(instances)
    images.append(np.zeros((size, size * n_empty)))
    for row in range(n_rows):
        rimages = images[row * images_per_row : (row + 1) * images_per_row]
        row_images.append(np.concatenate(rimages, axis=1))
    image = np.concatenate(row_images, axis=0)
    plt.imshow(image, cmap = mpl.cm.binary, **options)
    plt.axis("off")

#########################################################################################################################################
### Prepare MNIST dataset ###############################################################################################################
#########################################################################################################################################
mnist = fetch_openml('mnist_784', version=1)
mnist.target = mnist.target.astype(np.uint8)

X = mnist['data']       # MNIST 데이터셋에서 데이터를 불러옴
y = mnist['target']     # MNIST 데이터셋에서 Target 값을 불러옴

X_train, X_test, y_train, y_test = train_test_split(X, y)   # 데이터셋을 Training Set과 Test Set으로 분리함

#########################################################################################################################################
### PCA (Principle Component Analysis) ##################################################################################################
#########################################################################################################################################
pca = PCA()             # 데이터셋에 대한 Principle Component를 생성하는 PCA 객체 준비함
pca.fit(X_train)        # PCA를 통해 Training 데이터셋을 재구성함 (sklearn API는 표준화 과정을 내장하고 있음)
cumsum = np.cumsum(pca.explained_variance_ratio_)   # Principle Component의 분산의 누적합
d = np.argmax(cumsum >= 0.95) + 1                   # 데이터셋의 95%를 설명할 수 있는 Principle Component의 개수

print(d)    # 데이터셋의 95%를 설명할 수 있는 Principle Component의 개수를 출력함

plt.figure(figsize=(6,4))
plt.plot(cumsum, linewidth=3)               # Principle Component의 분산의 누적합을 그래프로 그림
plt.axis([0, 400, 0, 1])
plt.xlabel("Dimensions")
plt.ylabel("Explained Variance")

# 95%의 설명력 지점을 그래프로 표현함
plt.plot([d, d], [0, 0.95], "k:")
plt.plot([0, d], [0.95, 0.95], "k:")
plt.plot(d, 0.95, "ko")
plt.annotate("Elbow", xy=(65, 0.85), xytext=(70, 0.7),
             arrowprops=dict(arrowstyle="->"), fontsize=16)     # 95%의 설명력 지점을 그래프상 표현함

plt.grid(True)
save_fig("explained_variance_plot")
plt.show()

pca = PCA(n_components=0.95)                    # 95% 설명력을 가지는 Principle Component를 생성할 수 있는 PCA 객체 준비
X_reduced = pca.fit_transform(X_train)          # 95% 설명력의 Principle Component로 데이터셋을 재구성함 (sklearn API는 표준화를 내장함)

print(np.sum(pca.explained_variance_ratio_))    # 현재 사용하는 Principle Component의 분산 비율을 출력함

pca = PCA(n_components=154)                     # 154개의 상위 Principle Component를 생성할 수 있는 PCA 객체 준비
X_reduced = pca.fit_transform(X_train)          # 154개의 상위 Principle Component로 데이터셋을 재구성함 (sklearn API는 표준화를 내장함)
X_recovered = pca.inverse_transform(X_reduced)  # 재구성된 데이터셋을 원본 Feature 공간으로 재투영/복구시킴

X_reduced_pca = X_reduced   # PCA 기반 재구성된 데이터셋을 저장함

# 원본 데이터셋과 PCA에 의해 재구성된 데이터셋 간의 비교 그래프 출력
plt.figure(figsize=(7, 4))
plt.subplot(121)
plot_digits(X_train[::2100])                # MNIST Training 데이터셋을 전시함
plt.title("Original", fontsize=16)
plt.subplot(122)
plot_digits(X_recovered[::2100])            # PCA로 재투영된 Training 데이터셋을 전시함
plt.title("Compressed", fontsize=16)

save_fig("mnist_compression_plot")
plt.show()

#########################################################################################################################################
### IPCA (Incremental Principle Component Analysis) #####################################################################################
#########################################################################################################################################
n_batches = 100

inc_pca = IncrementalPCA(n_components=154)

for X_batch in np.array_split(X_train, n_batches):
    print('.', end='')
    inc_pca.partial_fit(X_batch)
print()

X_reduced_inc_pca = inc_pca.transform(X_train)

X_recovered_inc_pca = inc_pca.inverse_transform(X_reduced_inc_pca)

# 원본 데이터셋과 IPCA에 의해 재구성된 데이터셋 간의 비교 그래프 출력
plt.figure(figsize=(7, 4))
plt.subplot(121)
plot_digits(X_train[::2100])
plt.subplot(122)
plot_digits(X_recovered_inc_pca[::2100])
plt.tight_layout()
plt.show()

#########################################################################################################################################
### Comparison between PCA and IPCA #####################################################################################################
#########################################################################################################################################
print(np.allclose(pca.mean_, inc_pca.mean_))           # PCA 기반 데이터셋의 평균과 IPCA 기반 데이터셋의 평균을 비교함
print(np.allclose(X_reduced_pca, X_reduced_inc_pca))   # PCA 기반 데이터셋 구성과 IPCA 기반 데이터셋의 구성을 비교함
```



설명력 95% 지점을 도달하는 PCA Feature 개수 : 154

Original MNIST **PCA-Compressed MNIST**



Original MNIST **IPCA-Compressed MNIST**