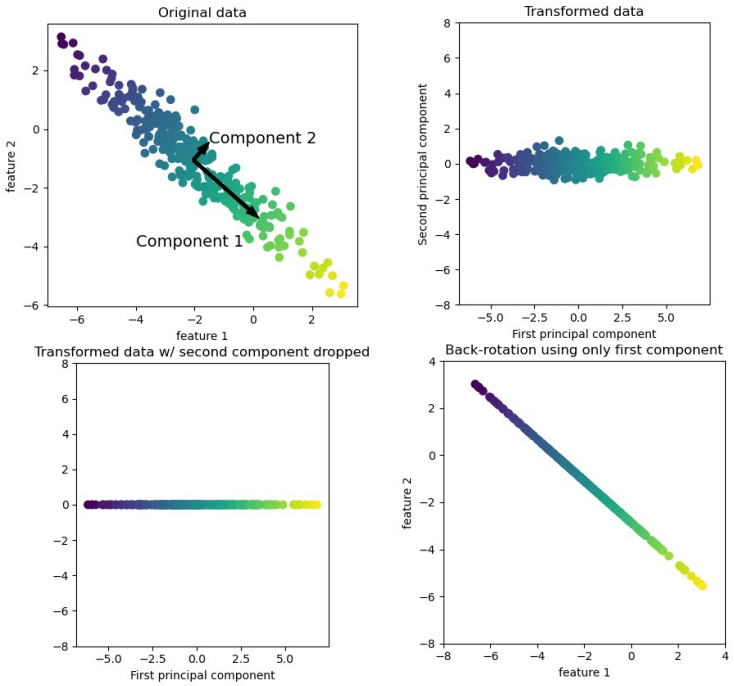
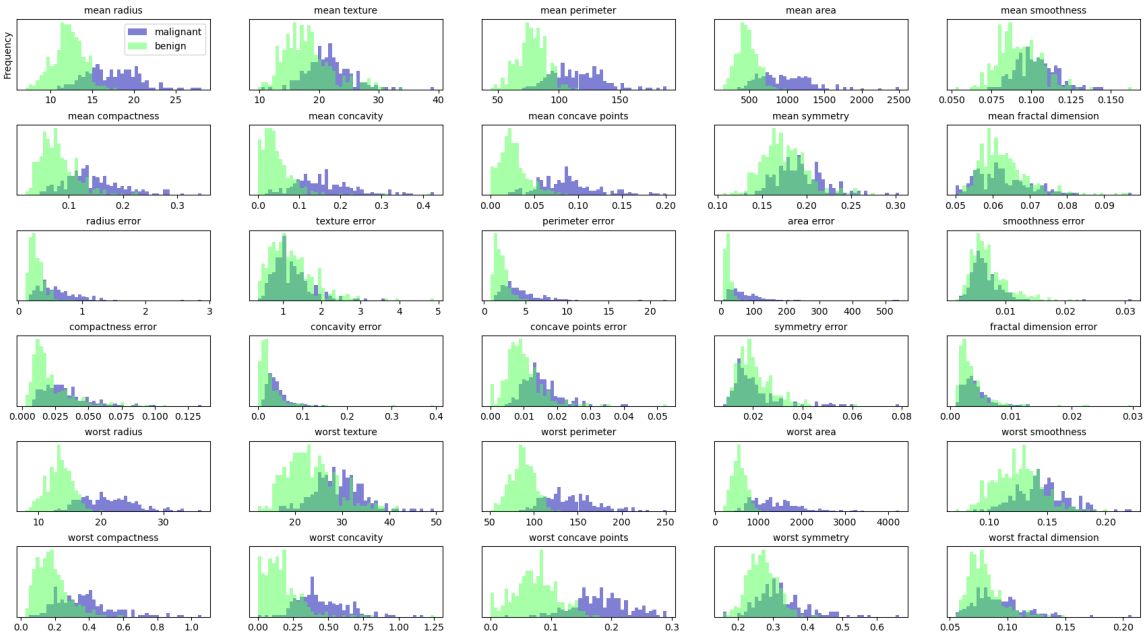
Q2-1. PCA of Breast Cancer Dataset

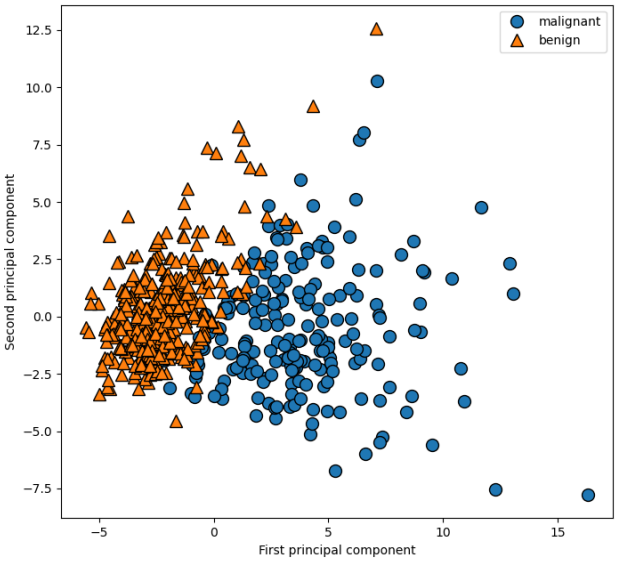
```
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import mglearn

from sklearn.datasets import load_breast_cancer

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# PCA 연산 과정 전시
mglearn.plots.plot_pca_illustration()
plt.show()

########################################################################################################################################################
### Dataset Preparation and Analysis ###################################################################################################################
########################################################################################################################################################
# 유방암 데이터셋 준비
cancer = load_breast_cancer()

fig, axes = plt.subplots(6, 5, figsize=(10, 20))

malignant = cancer.data[cancer.target == 0]     # 악성으로 분류된 데이터 준비
benign = cancer.data[cancer.target == 1]        # 양성으로 분류된 데이터 준비

# 데이터셋의 각 Feature 의 Historgram 을 그림
ax = axes.ravel()
for i in range(30):
    _, bins = np.histogram(cancer.data[:, i], bins=50)
    ax[i].hist(malignant[:, i], bins=bins, color=mglearn.cm3(0), alpha=.5)
    ax[i].hist(benign[:, i], bins=bins, color=mglearn.cm3(2), alpha=.5)
    ax[i].set_title(cancer.feature_names[i], fontsize=10)
    ax[i].set_yticks(())
    ax[0].set_ylabel("Frequency", fontsize=10)
    ax[0].legend(["malignant", "benign"], loc="best")

plt.subplots_adjust(left=0.05, right=0.95, top=0.95, bottom=0.05, hspace=0.5)
plt.show()

# 데이터셋 표준화
scaler = StandardScaler()                   # Standard Scaler 준비
scaler.fit(cancer.data)                     # 데이터셋에 대한 평균과 표준편차를 산출함
X_scaled = scaler.transform(cancer.data)    # 데이터셋을 표준화함

########################################################################################################################################################
### Principle Component Analysis of Breast Cancer Dataset ##############################################################################################
########################################################################################################################################################
pca = PCA(n_components=2)   # 2 개의 최상 Principle Component 를 생성하는 PCA 객체 준비
pca.fit(X_scaled)           # 표준화된 데이터셋에 대해 PCA 수행

X_pca = pca.transform(X_scaled)     # Principle Component 를 기반으로 데이터셋을 재구성함

print('Original shape : {}'.format(str(X_scaled.shape)))    # 원본 데이터의 형태를 출력함
print('Reduced shape : {}'.format(str(X_pca.shape)))        # PCA 기반으로 재구성된 데이터의 형태를 출력함

########################################################################################################################################################
### Result Plotting ####################################################################################################################################
########################################################################################################################################################

# 재구성된 데이터셋을 그래프로 그림
plt.figure(figsize=(8,8))
mglearn.discrete_scatter(X_pca[:,0], X_pca[:,1],cancer.target)
plt.legend(["malignant", "benign"], loc="best")
plt.gca().set_aspect("equal")
plt.xlabel("First principal component")
plt.ylabel("Second principal component")
plt.show()

print('PCA shape : ', pca.components_.shape)    # Principle Component 의 형태를 출력함

print('PCA components : ', pca.components_)     # Principle Component 를 출력함

# Principle Component 를 그래프로 그림
plt.matshow(pca.components_, cmap='viridis')
plt.yticks([0, 1], ["First component", "Second component"])
plt.colorbar()
plt.xticks(range(len(cancer.feature_names)), cancer.feature_names, rotation=60, ha='left')
plt.xlabel("Feature")
plt.ylabel("Principal components")
plt.show()
```



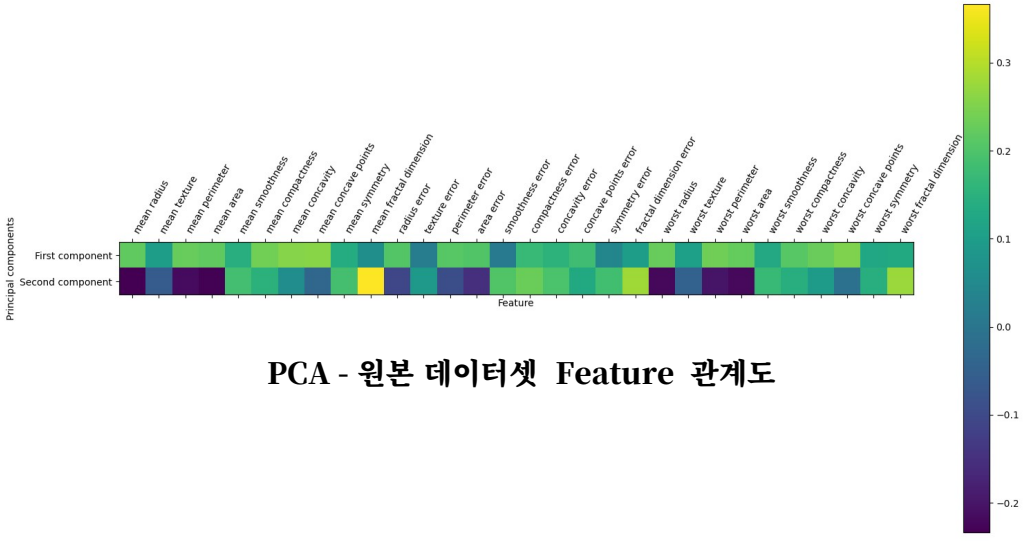
PCA 연산 과정



유방암 데이터셋의 각 Feature 의 Class 별 Histogram



PCA 로 재구성된 유방암 데이터셋 분포



PCA - 원본 데이터셋 Feature 관계도