PRACTICAL 4

Name:	Harsh Shah	**Semester:**	VII	**Division:**	6
Roll No.:	21BCP359	**Date:**	13-08-24	**Batch:**	G11
Aim:	Understanding Feature Extraction in Datasets.				

Question 1

Dataset: iris.csv

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

df = pd.read_csv('./Iris.csv')

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

 # Splitting Features and Target

X = df.drop(['Species'], axis=1)

y = df['Species']

```
X.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2

```
y.head()
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

 # Standard Scaler

scaler = StandardScaler()

X_standardized = scaler.fit_transform(X)

X_standardized_df = pd.DataFrame(X_standardized, columns=X.columns)

```
X_standardized_df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	-1.720542	-0.900681	1.032057	-1.341272	-1.312977
1	-1.697448	-1.143017	-0.124958	-1.341272	-1.312977
2	-1.674353	-1.385353	0.337848	-1.398138	-1.312977
3	-1.651258	-1.506521	0.106445	-1.284407	-1.312977
4	-1.628164	-1.021849	1.263460	-1.341272	-1.312977

```
X_standardized_df.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	1.500000e+02	1.500000e+02	1.500000e+02	1.500000e+02
mean	0.000000	-4.736952e-16	-6.631732e-16	3.315866e-16	-2.842171e-16
std	1.003350	1.003350e+00	1.003350e+00	1.003350e+00	1.003350e+00
min	-1.720542	-1.870024e+00	-2.438987e+00	-1.568735e+00	-1.444450e+00
25%	-0.860271	-9.006812e-01	-5.877635e-01	-1.227541e+00	-1.181504e+00
50%	0.000000	-5.250608e-02	-1.249576e-01	3.362659e-01	1.332259e-01
75%	0.860271	6.745011e-01	5.692513e-01	7.627586e-01	7.905908e-01
max	1.720542	2.492019e+00	3.114684e+00	1.786341e+00	1.710902e+00

 # Principle Component Analysis

pca = PCA(n_components=2)

principal_components = pca.fit_transform(X_standardized)

principal_df = pd.DataFrame(principal_components, columns=['PC1', 'PC2'])

final_df = pd.concat([principal_df, y], axis=1)

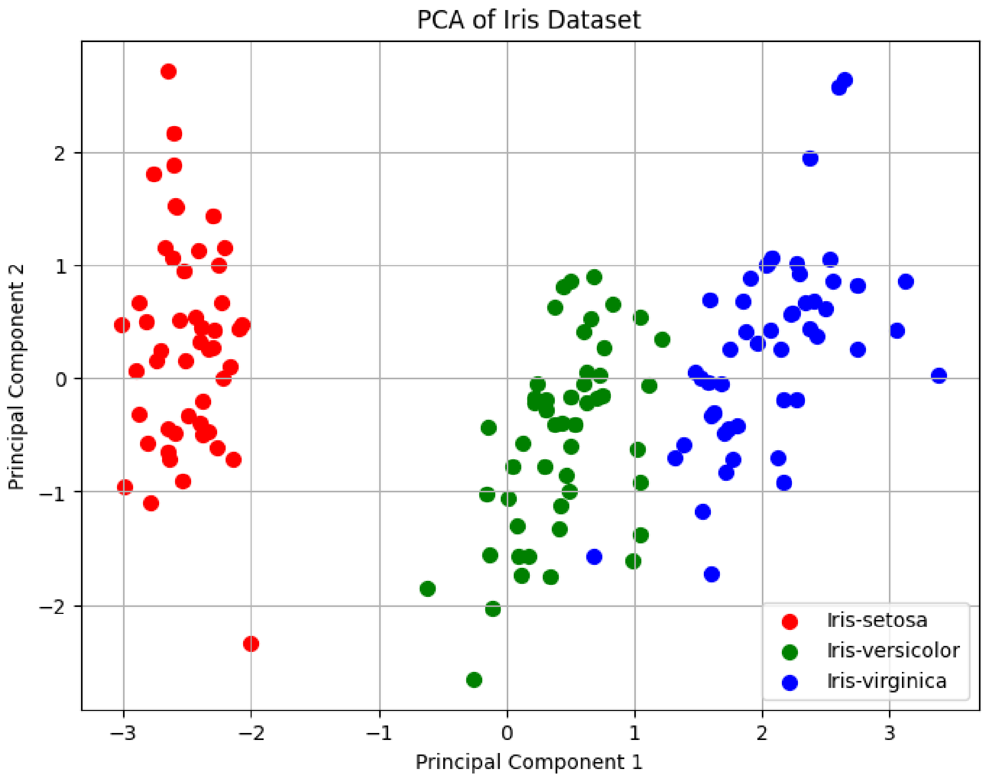
```
final_df.head()
```

	PC1	PC2	Species
0	-2.816339	0.506051	Iris-setosa
1	-2.645527	-0.651799	Iris-setosa
2	-2.879481	-0.321036	Iris-setosa
3	-2.810934	-0.577363	Iris-setosa
4	-2.879884	0.670468	Iris-setosa

 # Plot

```
plt.figure(figsize=(8, 6))
colors = ['red', 'green', 'blue']
species_names = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
for species, color in zip(species_names, colors):
    indices_to_keep = final_df['Species'] == species
    plt.scatter(final_df.loc[indices_to_keep, 'PC1'],
                final_df.loc[indices_to_keep, 'PC2'],
                c=color, s=50, label=species)

# Add labels and title
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Iris Dataset')
plt.legend()
plt.grid()
```



Question 2

Dataset: wine.csv

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

df = pd.read_csv('./wine_data.csv')

```
df.head()
```

	class_label	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	OD315_of_d
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	

X = df.drop(['class_label'], axis=1)

y = df['class_label']

 # Standardization

scaler = StandardScaler()

X_standardized = scaler.fit_transform(X)

X_standardized_df = pd.DataFrame(X_standardized, columns=X.columns)

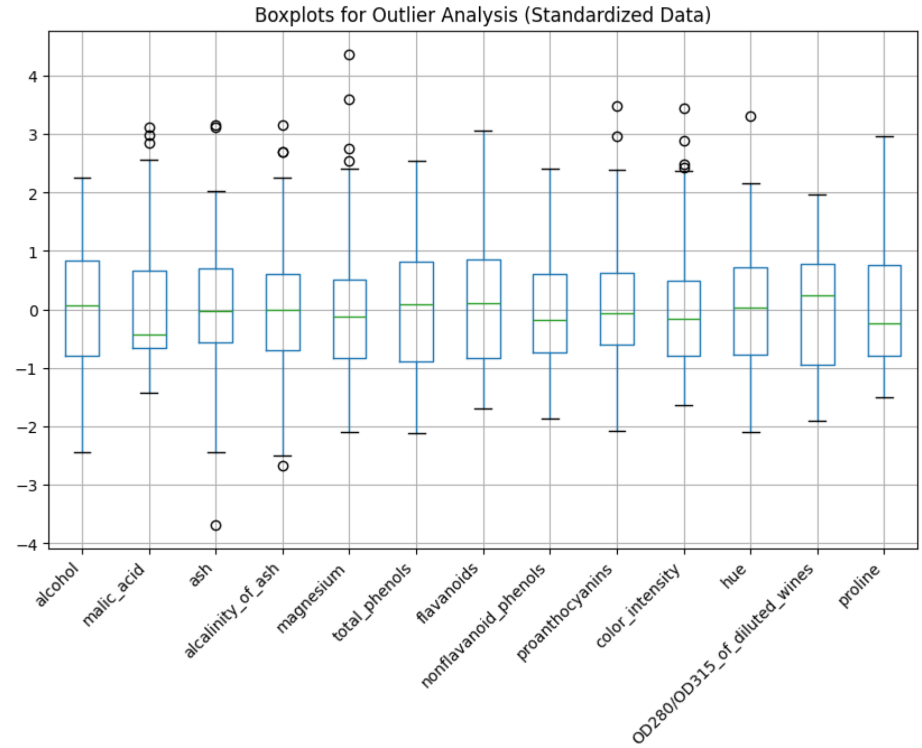
plt.figure(figsize=(10, 6))

X_standardized_df.boxplot()

plt.xticks(rotation=45, ha='right')

plt.title('Boxplots for Outlier Analysis (Standardized Data)')

plt.show()



 # Covariance Matrix

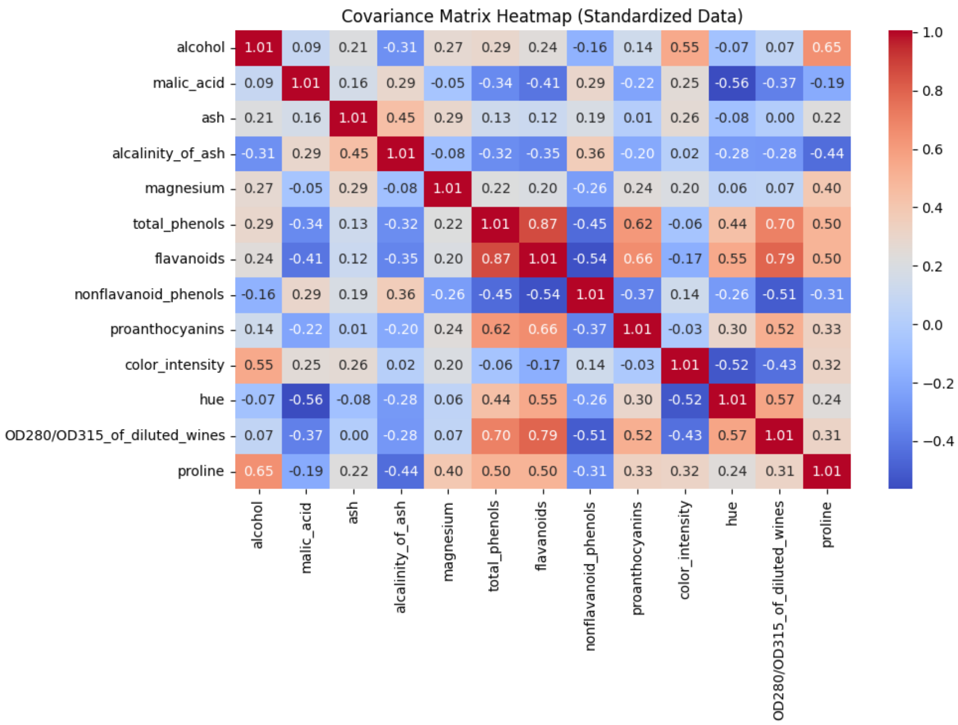
cov_matrix_standardized = pd.DataFrame(X_standardized, columns=X.columns).cov()

plt.figure(figsize=(10, 6))

sns.heatmap(cov_matrix_standardized, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Covariance Matrix Heatmap (Standardized Data)')

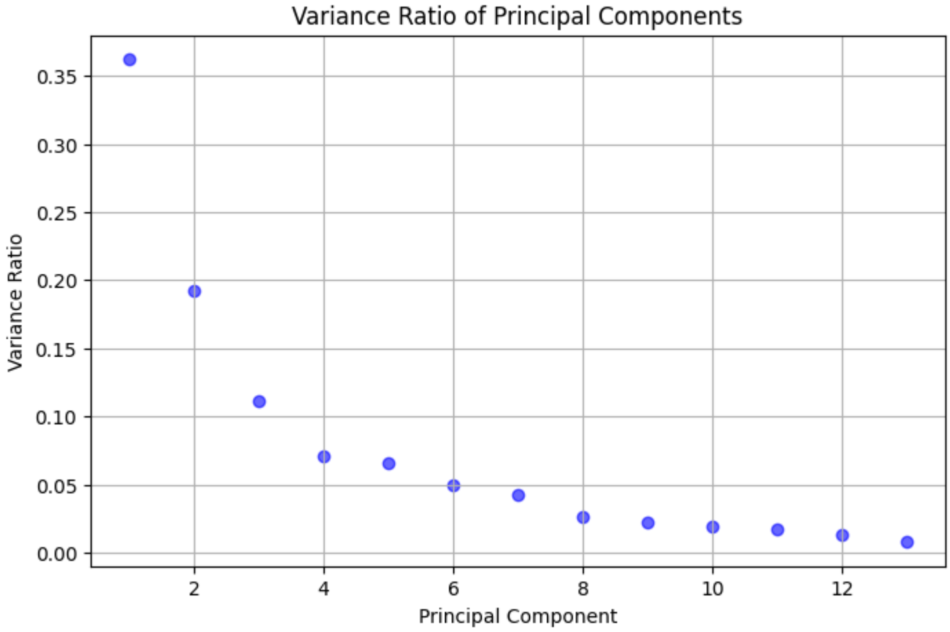
plt.show()



 # PCA without specifying components

```
pca = PCA(n_components=None)
pca.fit(X_standardized)


plt.figure(figsize=(8, 5))
plt.scatter(range(1, len(pca.explained_variance_ratio_) + 1), pca.explained_variance_ratio_, label='Variance Ratio', color='blue', alpha=0.6)
# plt.plot(range(1, len(pca.explained_variance_ratio_) + 1), pca.explained_variance_ratio_)
plt.xlabel('Principal Component')
plt.ylabel('Variance Ratio')
plt.title('Variance Ratio of Principal Components')
plt.grid()
plt.show()
```



 # PCA with 2 components

```
pca_2d = PCA(n_components=2)
principal_components = pca_2d.fit_transform(X_standardized)


principal_df = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2'])
final_df = pd.concat([principal_df, y.reset_index(drop=True)], axis=1)
```

```
plt.figure(figsize=(10, 6))

colors = ['red', 'green', 'blue']
for label, color in zip(df['class_label'].unique(), colors):
    indices_to_keep = final_df['class_label'] == label
    plt.scatter(final_df.loc[indices_to_keep, 'PC1'],
                final_df.loc[indices_to_keep, 'PC2'],
                c=color, s=50, label=label, alpha=0.6)

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Wine Dataset (2 Components)')
plt.legend()
plt.grid()
plt.show()
```