**Question**

Write a Python program to demonstrate data preprocessing steps: handling missing values, encoding categorical data, and feature scaling.

1. Load a sample dataset using pandas (e.g., Iris and Wine datasets).

2. Plot the distribution of a feature using matplotlib.pyplot.hist().

3. Create scatter plots to understand relationships between features using seaborn.scatterplot().

4. Use a correlation heatmap to find the relationship between multiple features with seaborn.heatmap().

**Iris Dataset**

**Source Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder, StandardScaler

# Loading dataset

dataset\_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'class']

iris = pd.read\_csv(dataset\_url, header=None, names=columns)

print(iris)

# Imputation not needed due to no missing values

# Label Encoding the classes (textual data -> numerical)

label\_encoder = LabelEncoder()

iris['class'] = label\_encoder.fit\_transform(iris['class'])

# Feature scaling

scaler = StandardScaler()

iris[['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width']] = scaler.fit\_transform(

iris[['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width']]

)

# Plotting the frequency distribution of Sepal Length

plt.figure(figsize=(6, 4))

plt.hist(iris['sepal\_length'], bins=20, color='blue', alpha=0.7, edgecolor='black')

plt.title('Distribution of Sepal Length', fontsize=16)

plt.xlabel('Sepal Length', fontsize=12)

plt.ylabel('Frequency', fontsize=12)

plt.grid(True)

plt.show()

# Scatter plot relating Sepal Length and Petal Length

plt.figure(figsize=(6, 4))

sns.scatterplot(data=iris, x='sepal\_length', y='petal\_length', hue='class', palette='Set2')

plt.title('Scatter Plot: Sepal Length vs Petal Length', fontsize=16)

plt.xlabel('Sepal Length', fontsize=12)

plt.ylabel('Petal Length', fontsize=12)

plt.legend(title='Class')

plt.show()

# Correlation heatmap

plt.figure(figsize=(8, 6))

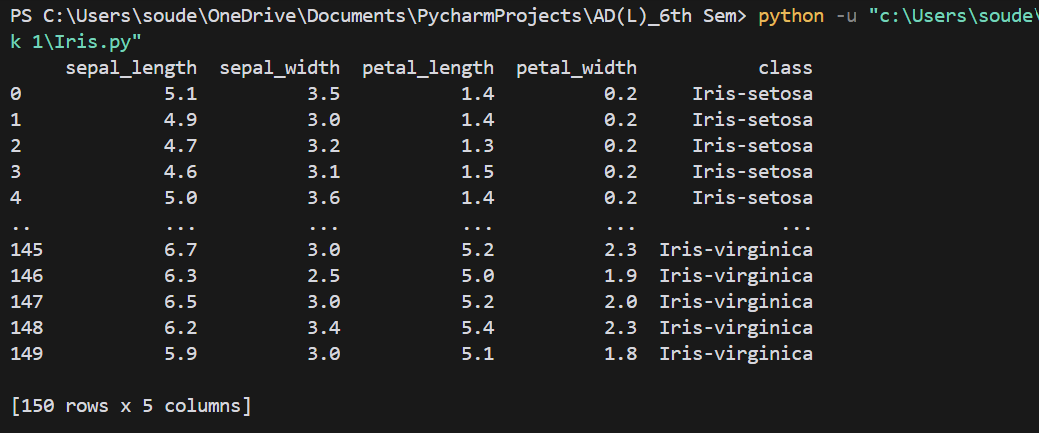
corr\_matrix = iris.corr()

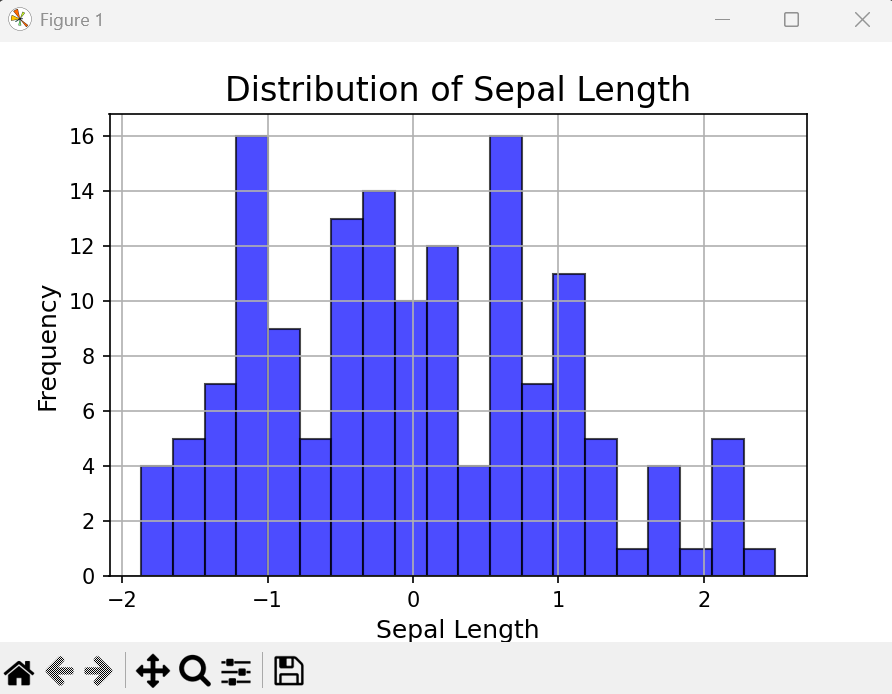
sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Heatmap', fontsize=16)

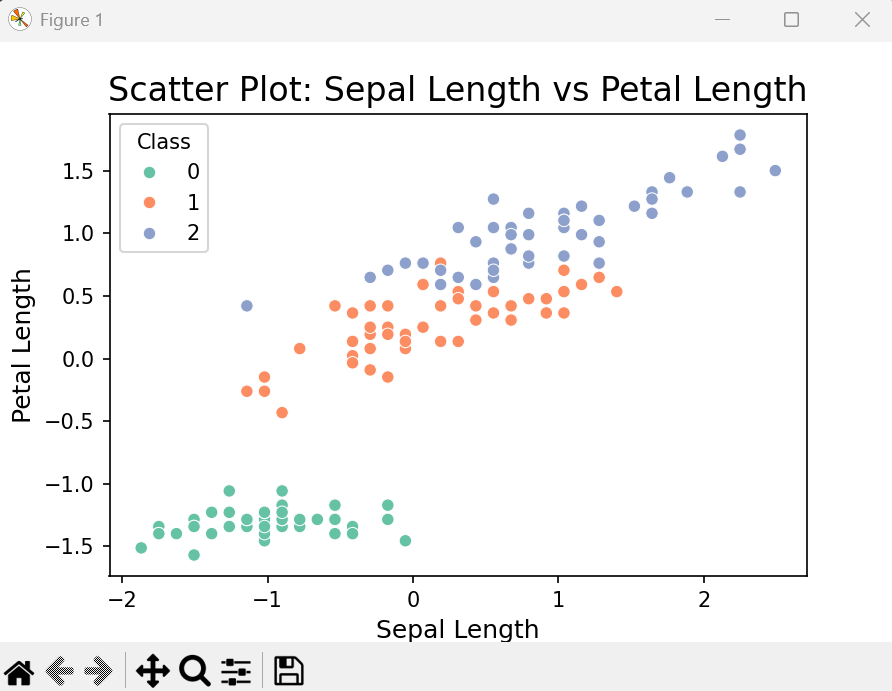
plt.show()

**Output**

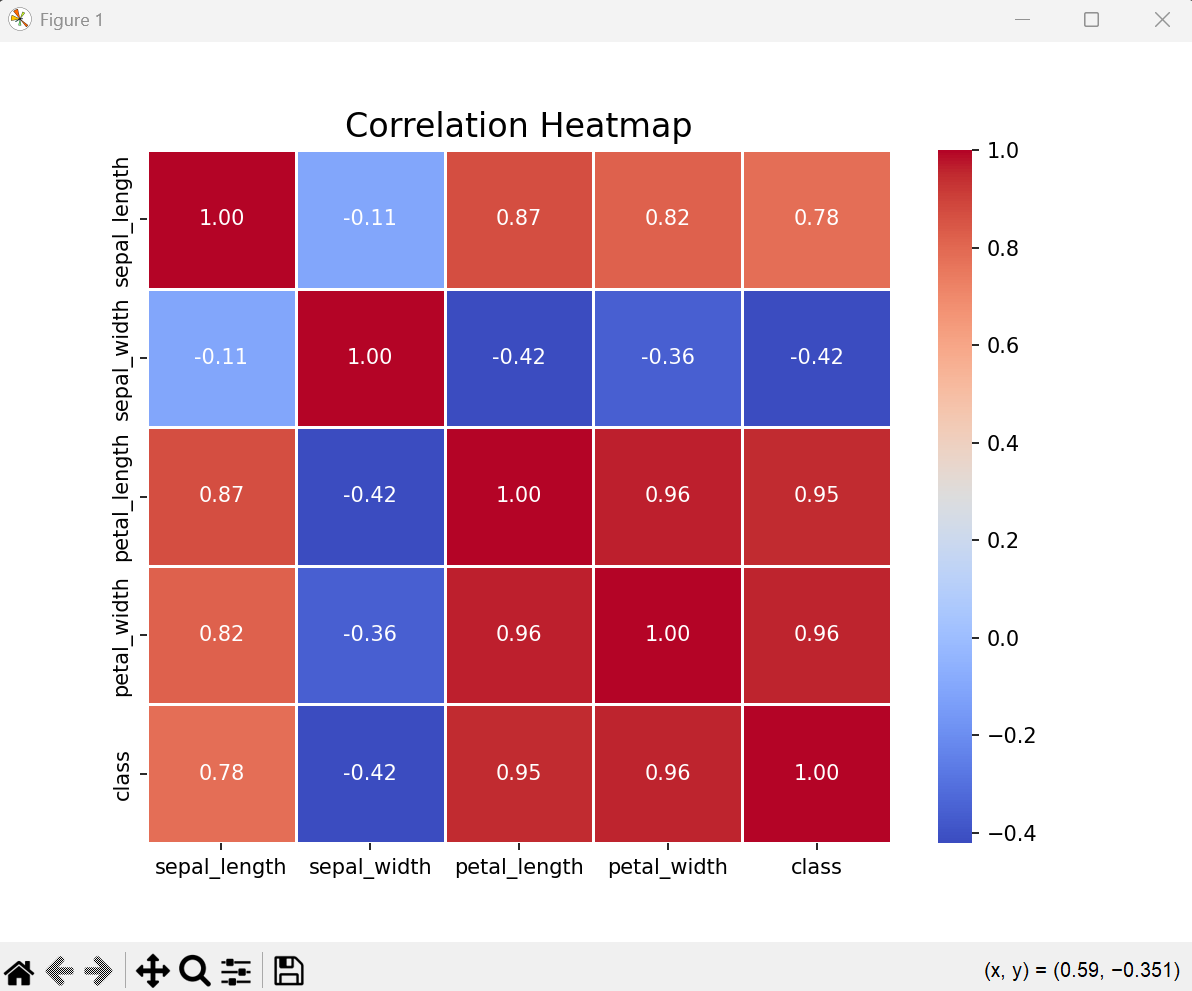
*Terminal*

******

*Plot of distribution of Sepal Length using Pyplot*

******

*Scatter plot of Sepal Length vs Petal Length using Seaborn*

*Feature Correlation Heatmap*

**Wine Dataset**

**Source Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder, StandardScaler

# Load the Wine dataset

dataset\_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"

columns = ['class', 'alcohol', 'malic\_acid', 'ash', 'alcalinity\_of\_ash', 'magnesium', 'total\_phenols', 'flavanoids',

'nonflavanoid\_phenols', 'proanthocyanins', 'color\_intensity', 'hue', 'od280/od315\_of\_diluted\_wines', 'proline']

wine = pd.read\_csv(dataset\_url, header=None, names=columns)

print(wine.head())

# Handling missing values

print(wine.isnull().sum()) #Print the number of missing values

# No imputation needed due to no missing value

# Feature scaling

scaler = StandardScaler()

wine.iloc[:, 1:] = scaler.fit\_transform(wine.iloc[:, 1:])

# Plotting the distribution of a Alcohol

plt.figure(figsize=(6, 4))

plt.hist(wine['alcohol'], bins=20, color='blue', alpha=0.7, edgecolor='black')

plt.title('Distribution of Alcohol', fontsize=16)

plt.xlabel('Alcohol (scaled)', fontsize=12)

plt.ylabel('Frequency', fontsize=12)

plt.grid(True)

plt.show()

# Scatter plot of *Alcohol vs Color Intensity*

plt.figure(figsize=(6, 4))

sns.scatterplot(data=wine, x='alcohol', y='color\_intensity', hue='class', palette='Set2')

plt.title('Scatter Plot: Alcohol vs Color Intensity', fontsize=16)

plt.xlabel('Alcohol (scaled)', fontsize=12)

plt.ylabel('Color Intensity (scaled)', fontsize=12)

plt.legend(title='Class')

plt.show()

# Correlation heatmap

plt.figure(figsize=(8, 6))

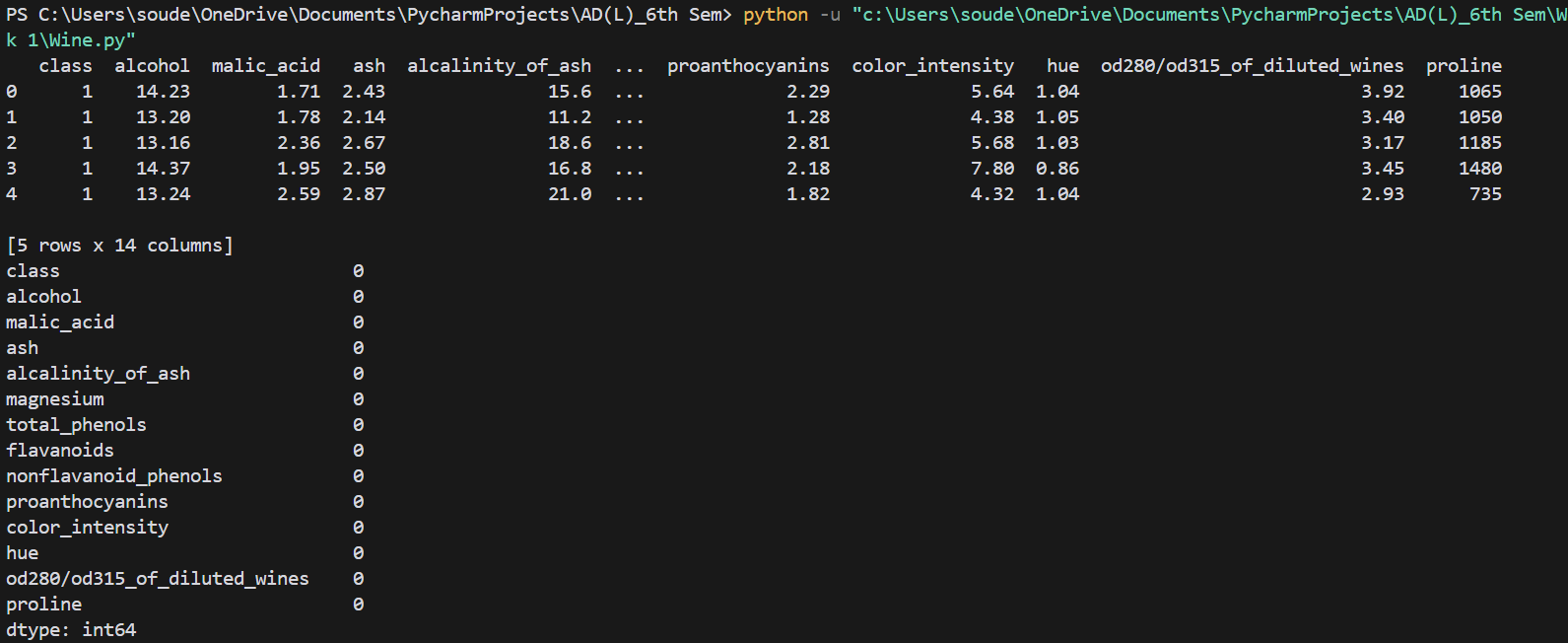
corr\_matrix = wine.iloc[:, 1:].corr()

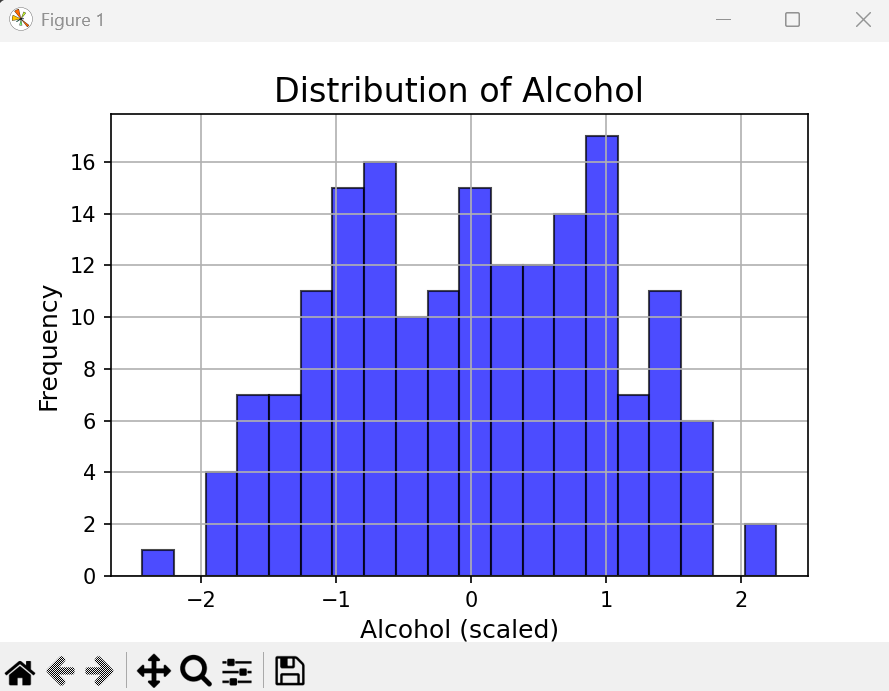
sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Heatmap', fontsize=16)

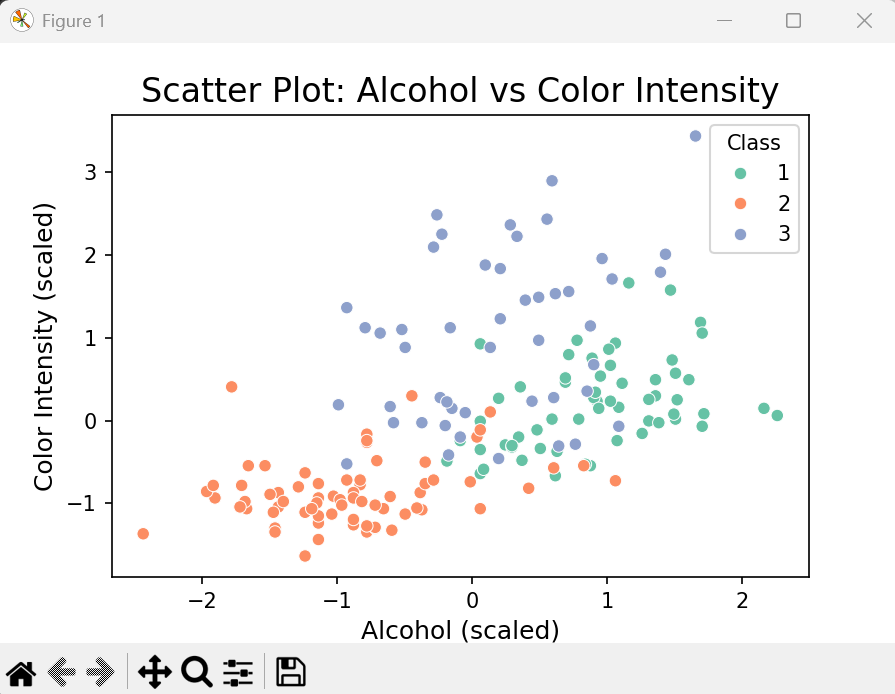
plt.show()

**Output**

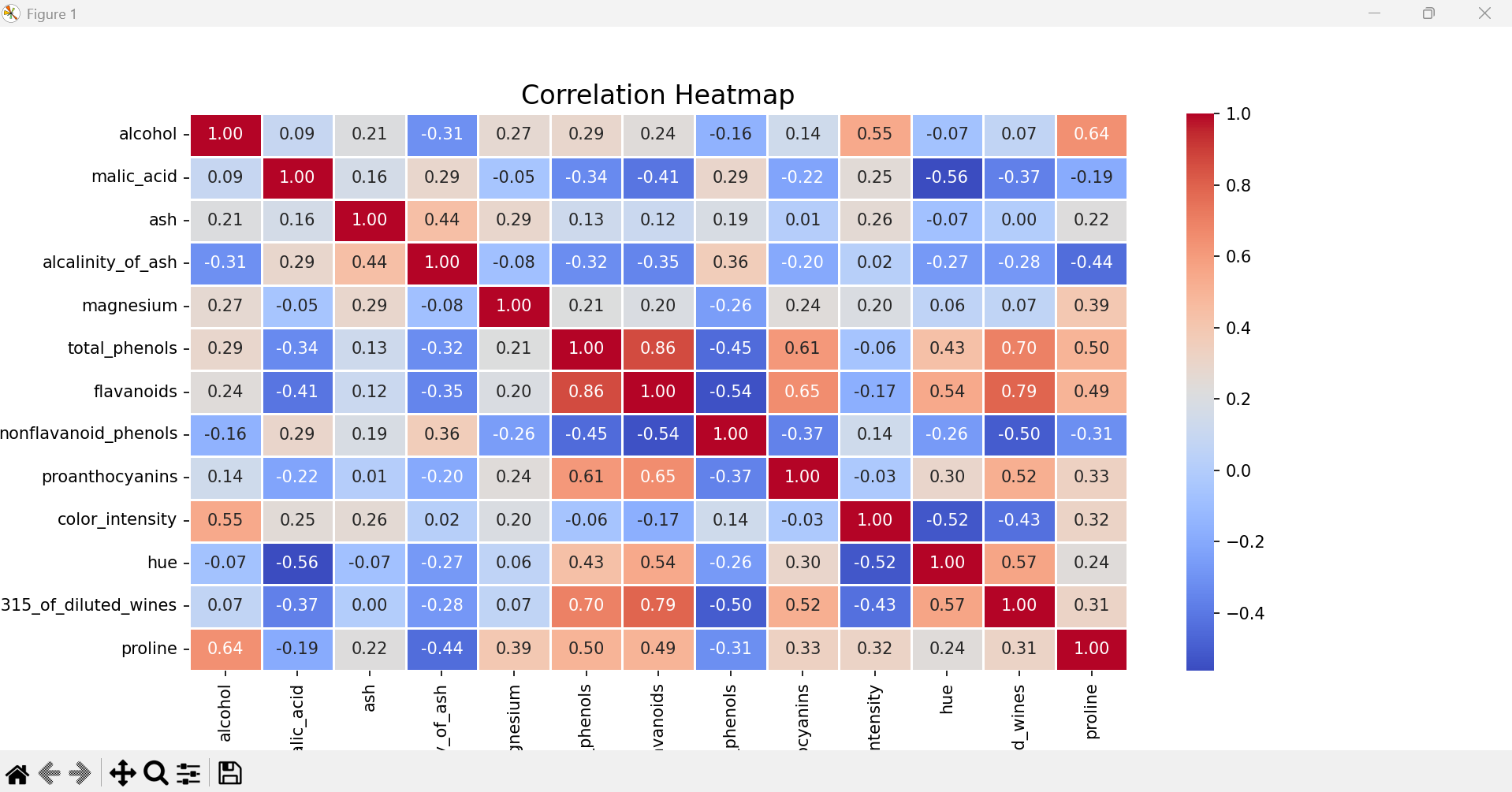
*Terminal*

******

*Plot of distribution of Alcohol using Pyplot*

******

*Scatter plot of Alcohol vs Color Intensity using Seaborn*

*Feature Correlation Heatmap*