```
In [ ]: pip install otter-grader scikit-learn
```

```
In [ ]: !pip install seaborn
```

```
In [10]: !pip list | grep seaborn
```

```
seaborn                           0.13.2
```

```
In [ ]: !pip install --upgrade --force-reinstall seaborn
```

```
In [9]: !pip list | grep seaborn
```

```
seaborn                           0.13.2
```

```
In [1]: import sys
        print(sys.executable)
```

```
/Library/Developer/CommandLineTools/usr/bin/python3
```

```
In [2]: import site
        print(site.getsitepackages())
```

```
['/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/V
ersions/3.9/lib/python3.9/site-packages', '/Library/Python/3.9/site-package
s', '/AppleInternal/Library/Python/3.9/site-packages', '/AppleInternal/Test
s/Python/3.9/site-packages']
```

```
In [ ]: !python3 -m pip install seaborn
```

```
In [4]: !find /Library/Python/3.9/site-packages -name "seaborn*"
```

```
find: /Library/Python/3.9/site-packages: No such file or directory
```

```
In [37]: # Initialize Otter
         import otter
         grader = otter.Notebook("1-235e6.ipynb")
```

K-means Clustering and Dimensionality Reduction on Wine Dataset

Introduction

In this problem set, we will revisit key clustering and dimensionality reduction using Principal Component Analysis (PCA) and k-Means on the Wine dataset. As you work through the notebook, please follow the sequence and address the questions embedded along the way. We'll also explore visualizing the results, interpreting them, and discussing their implications for a hypothetical wine retailer. Please keep your written answers to 300 words at max.

Let's begin by importing the necessary packages and loading the Wine dataset.

If you encounter any issues with missing packages, install them by running `%pip install <package_name>`, for example `%pip install matplotlib`.

In [38]:

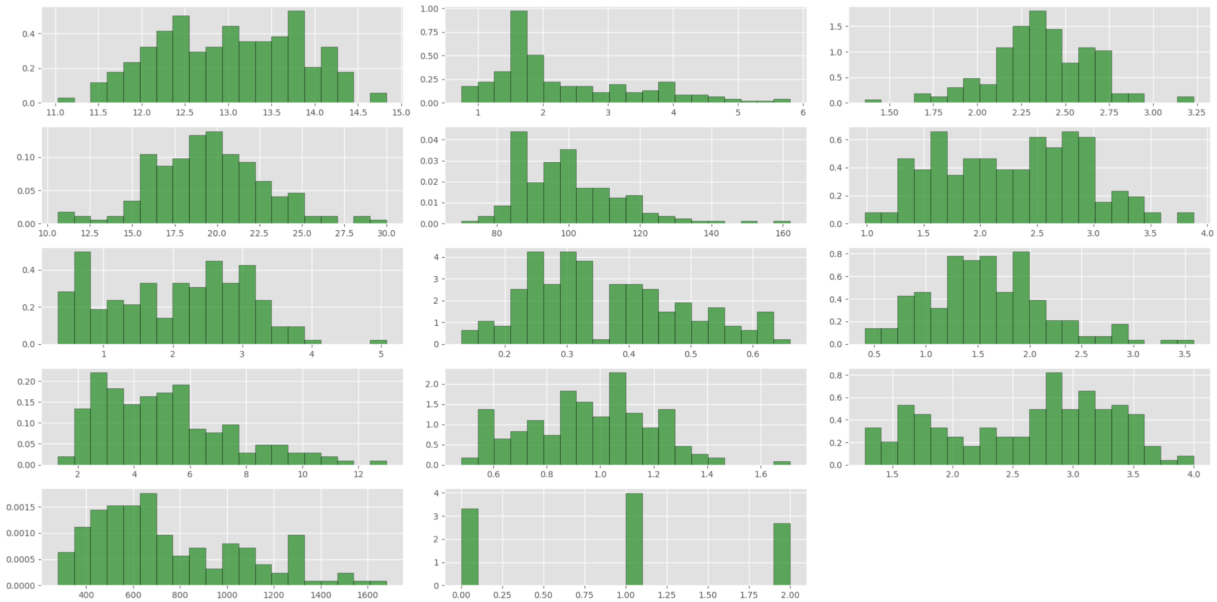
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score

%matplotlib inline
plt.style.use('ggplot')

# Load the Wine dataset
wine_data = load_wine()

df = pd.DataFrame(wine_data.data, columns=wine_data.feature_names)
df['label'] = wine_data.target
df.head()

print(wine_data.DESCR)
```

```
.. _wine_dataset:

Wine recognition dataset
------------------------

**Data Set Characteristics:**

:Number of Instances: 178
:Number of Attributes: 13 numeric, predictive attributes and the class
:Attribute Information:
    - Alcohol
    - Malic acid
    - Ash
    - Alcalinity of ash
    - Magnesium
    - Total phenols
    - Flavanoids
    - Nonflavanoid phenols
    - Proanthocyanins
    - Color intensity
    - Hue
    - OD280/OD315 of diluted wines
    - Proline
    - class:
        - class_0
        - class_1
        - class_2

:Summary Statistics:

============================= ==== ===== ======= =====
                                Min   Max   Mean     SD
============================= ==== ===== ======= =====
Alcohol:                      11.0  14.8    13.0   0.8
Malic Acid:                   0.74  5.80    2.34  1.12
Ash:                          1.36  3.23    2.36  0.27
Alcalinity of Ash:            10.6  30.0    19.5   3.3
Magnesium:                    70.0 162.0    99.7  14.3
Total Phenols:                0.98  3.88    2.29  0.63
Flavanoids:                   0.34  5.08    2.03  1.00
Nonflavanoid Phenols:         0.13  0.66    0.36  0.12
Proanthocyanins:              0.41  3.58    1.59  0.57
Colour Intensity:              1.3  13.0     5.1   2.3
Hue:                          0.48  1.71    0.96  0.23
OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71
Proline:                       278  1680     746   315
============================= ==== ===== ======= =====

:Missing Attribute Values: None
:Class Distribution: class_0 (59), class_1 (71), class_2 (48)
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
:Date: July, 1988

This is a copy of UCI ML Wine recognition datasets.
https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data
```

```
The data is the results of a chemical analysis of wines grown in the same
region in Italy by three different cultivators. There are thirteen different
measurements taken for different constituents found in the three types of
wine.

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies,
Via Brigata Salerno, 16147 Genoa, Italy.

Citation:

Lichman, M. (2013). UCI Machine Learning Repository
[https://archive.ics.uci.edu/ml]. Irvine, CA: University of California,
School of Information and Computer Science.

.. dropdown:: References

    (1) S. Aeberhard, D. Coomans and O. de Vel,
    Comparison of Classifiers in High Dimensional Settings,
    Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of
    Mathematics and Statistics, James Cook University of North Queensland.
    (Also submitted to Technometrics).

    The data was used with many others for comparing various
    classifiers. The classes are separable, though only RDA
    has achieved 100% correct classification.
    (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data))
    (All results using the leave-one-out technique)

    (2) S. Aeberhard, D. Coomans and O. de Vel,
    "THE CLASSIFICATION PERFORMANCE OF RDA"
    Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of
    Mathematics and Statistics, James Cook University of North Queensland.
    (Also submitted to Journal of Chemometrics).
```

In [39]:

```python
X_wine = pd.DataFrame(wine_data.data, columns=wine_data.feature_names)

features = df.columns.to_list()
plt.figure(figsize = (20, 10))
for i in range(0, len(features)):
    plt.subplot(5, 3, i+1)
    plt.hist(df[features[i]], bins=20, color='green', alpha=0.6, edgecolor='
    plt.tight_layout()
```



Before applying K-means clustering, we need to standardize the data to ensure all features contribute equally to the clustering process.

In [40]:
```
# Standardize the data
scaler = StandardScaler()
X_wine_scaled = scaler.fit_transform(X_wine)
```

In [41]:
```
X_wine
```

Out[41]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoid
0	14.23	1.71	2.43	15.6	127.0	2.80	3.0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.7
2	13.16	2.36	2.67	18.6	101.0	2.80	3.2
3	14.37	1.95	2.50	16.8	113.0	3.85	3.4
4	13.24	2.59	2.87	21.0	118.0	2.80	2.6
...	...	...	...	...	...	...	.
173	13.71	5.65	2.45	20.5	95.0	1.68	0.6
174	13.40	3.91	2.48	23.0	102.0	1.80	0.7
175	13.27	4.28	2.26	20.0	120.0	1.59	0.6
176	13.17	2.59	2.37	20.0	120.0	1.65	0.6
177	14.13	4.10	2.74	24.5	96.0	2.05	0.7

178 rows × 13 columns

K-means Clustering

K-means clustering is an unsupervised machine learning algorithm used to partition a dataset into $k$ clusters. Each observation belongs to the cluster with the nearest mean (centroid).

K-means clustering can be used in various scenarios, such as customer segmentation in marketing, anomaly detection or document clustering. In our case, we'll use it to group similar wines together based on their characteristics.

In [42]:

```python
# Perform K-means clustering with 3 clusters
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(X_wine_scaled)
```

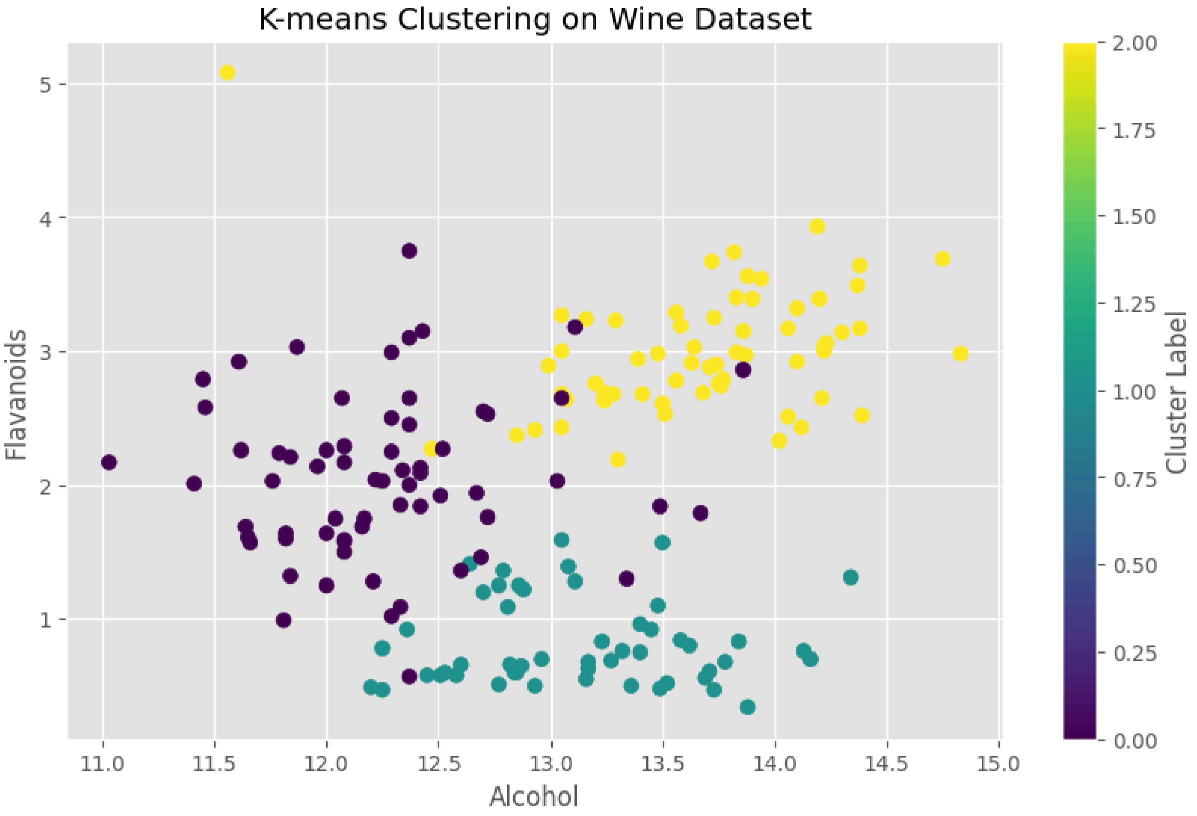
Question 1 (0.5 points)

Produce a scatterplot with a combination of two features from the dataset, colored to show the k-means clusters. Choose a combination of features which shows the three clusters as relatively distinct.

Hint: consider alcohol, malic acid, flavonoids and color intensity

In [43]:

```python
# Choose two features for visualization
feature_1 = 'alcohol'
feature_2 = 'flavanoids'


# Create a scatter plot for the clusters
plt.figure(figsize=(10, 6))
scatter = plt.scatter(X_wine[feature_1], X_wine[feature_2], c=kmeans_labels,
plt.title('K-means Clustering on Wine Dataset')
plt.xlabel(feature_1.capitalize())
plt.ylabel(feature_2.capitalize())
plt.colorbar(scatter, label='Cluster Label')
plt.show()
```



Question 2 (2 points)

Based on the plot, describe the characteristics of each cluster in terms of your features. How well-separated are the clusters?

The clusters are reasonably well-separated, especially the Green from the other two. There is a clear distinction between the low flavanoid wines and the others. However, there is some overlap between the purple cluster and the yellow one, particularly in the region where alcohol content is around 12.7-14% and flavanoids are around 1.5-3. This overlap suggests that some wines in these clusters share similar characteristics and may be harder to distinguish based solely on these two features indicating that additional features might be needed to fully distinguish between these wine types. Nonetheless, the overall pattern shows distinct groupings, suggesting that the clustering has captured meaningful differences in the wine characteristics.

Principal Component Analysis (PCA)

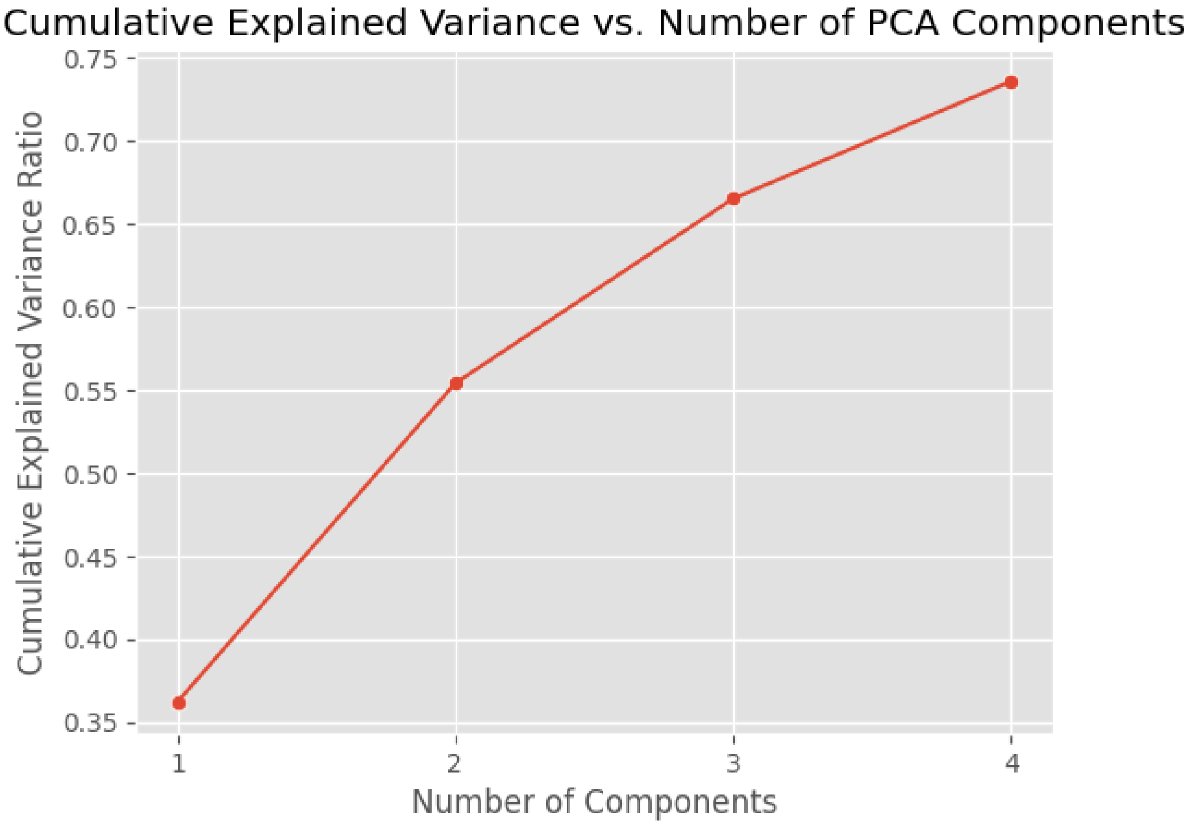
Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms data into a new coordinate system. PCA takes a complex, multi-dimensional dataset and finds a simpler way to represent it that still keeps key information. It's often used to visualize high-dimensional data in 2D or 3D plots, reduce noise in data, compress data while minimizing information loss, or prepare data for machine learning algorithms.

Let's apply PCA to our wine dataset and reduce it to 4 dimensions.

In [44]:
```
import seaborn as sns

n_components = 4
pca = PCA(n_components=n_components, random_state=211)
X_pca = pca.fit_transform(X_wine_scaled)

explained_variance = pca.explained_variance_ratio_
print(f'Explained variance by PCA components: {explained_variance}')


component_nums = list(range(1, n_components+1))
sns.lineplot(x=component_nums, y=np.cumsum(explained_variance))
ax = sns.scatterplot(x=component_nums, y=np.cumsum(explained_variance))
ax.set_xlabel('Number of Components')
ax.set_ylabel('Cumulative Explained Variance Ratio')
ax.set_title('Cumulative Explained Variance vs. Number of PCA Components')
ax.set(xticks= component_nums)
```

```
Explained variance by PCA components: [0.36198848 0.1920749  0.11123631 0.07
06903 ]
```

Out[44]:
```
[[<matplotlib.axis.XTick at 0x120137b50>,
  <matplotlib.axis.XTick at 0x120137b20>,
  <matplotlib.axis.XTick at 0x11d562e50>,
  <matplotlib.axis.XTick at 0x11d7a41f0>]]
```



Question 3 (0.5 points)

What is the smallest number of components needed to account for 50% of the variance?

In [45]:

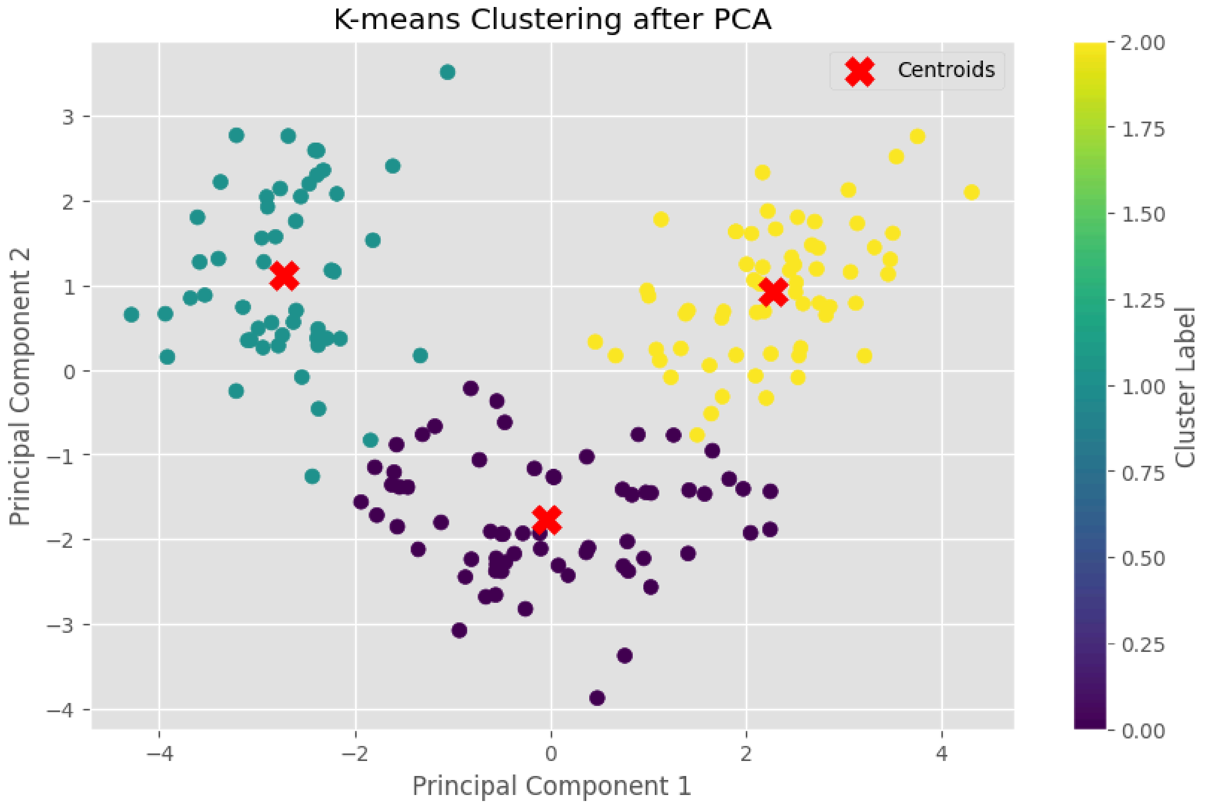
```
n_components_which_account_for_50_percent_variance = 1.6
```

K-means Clustering on PCA-reduced Data

Now, let's perform K-means clustering on the PCA-reduced dataset and visualize the results.

In [46]:

```
# Perform K-means on the reduced PCA dataset
kmeans_pca = KMeans(n_clusters=3, random_state=42)
kmeans_labels_pca = kmeans_pca.fit_predict(X_pca)

# Scatter plot of the clusters in the PCA-reduced space
plt.figure(figsize=(10, 6))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_labels_pca, cmap='v
plt.title('K-means Clustering after PCA')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(scatter, label='Cluster Label')

# Get the centroids of the clusters after PCA
centroids_pca = kmeans_pca.cluster_centers_

# Plot the centroids on the scatter plot. This will help you visualise how s
plt.scatter(centroids_pca[:, 0], centroids_pca[:, 1], s=200, c='red', marker
plt.legend()
plt.show()
```



We're also going to compare the quality of clustering before and after PCA using the silhouette score:

In [47]:

```
# K-means on the original dataset (without PCA)
kmeans_original = KMeans(n_clusters=3, random_state=42)
kmeans_labels_original = kmeans_original.fit_predict(X_wine_scaled)

# Compare the clustering results
silhouette_pca = silhouette_score(X_pca, kmeans_labels_pca)
silhouette_original = silhouette_score(X_wine_scaled, kmeans_labels_original

print(f'Silhouette Score after PCA: {silhouette_pca:.4f}')
print(f'Silhouette Score on Original Data: {silhouette_original:.4f}')
```

```
Silhouette Score after PCA: 0.4051
Silhouette Score on Original Data: 0.2849
```

Question 3 (1 points)

Compare the clustering results before and after PCA. How do they differ?

What new observations can you make from the PCA-based clustering?

Do the silhouette scores agree with what you see on the plots?

there seems to be a better separation of the clusters after PCA, as evidenced by the silhouette score increase. Also, it is ontable how seems the colors after the PCA didnt cross with each other making it visually easier to distinguish the clusters. The silhouette scores agree with what we see on the plots, with the PCA-based clustering having a higher silhouette score, indicating that it is a better clustering.

Interpreting PCA Components

Question 4a (1.5 points)

After performing PCA on the Wine dataset, your goal is to display the top 5 contributing features for Principal Component 1 (PC1) and Principal Component 2 (PC2). Complete the Python program to extract the features that contribute most to each of these principal components.

In [48]:

```
# Get the PCA component loadings
pca_component_loadings = pd.DataFrame(pca.components_, columns=wine_data.fea
pca_component_loadings
```

Out[48]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flav
0	0.144329	-0.245188	-0.002051	-0.239320	0.141992	0.394661	0.4
1	0.483652	0.224931	0.316069	-0.010591	0.299634	0.065040	-0.0
2	-0.207383	0.089013	0.626224	0.612080	0.130757	0.146179	0.
3	-0.017856	0.536890	-0.214176	0.060859	-0.351797	0.198068	0.

In [49]:

```
#  Extract the first PC loadings from the DataFrame by indexing it
pc_1 = pca_component_loadings.iloc[0]

# Sort the PC and take the first 5 values
top_5_features_pc_1 =  pca_component_loadings.iloc[0].nlargest(5).index.toli

# Do the same thing for the second PC
top_5_features_pc_2 = pca_component_loadings.iloc[1].nlargest(5).index.tolis
print(top_5_features_pc_1)
print(top_5_features_pc_2)
```

```
['flavanoids', 'total_phenols', 'od280/od315_of_diluted_wines', 'proanthocya
nins', 'hue']
['color_intensity', 'alcohol', 'proline', 'ash', 'magnesium']
```

Question 4b (0.5 points)

What do the positive and negative contributions signify?

Positive contributions: Features with positive values contribute to increasing the value of that principal component. A larger positive value means the feature has a stronger positive influence on that component. Negative contributions: Features with negative values contribute to decreasing the value of that principal component. A larger negative value means the feature has a stronger negative influence on that component.*Type your answer here, replacing this text.*

Application to Wine Retail

Question 5 (1 points)

Based on the K-means clustering results and the PCA analysis, identify two features you would choose to create distinct sections in your wine store. Justify your selection and explain how you would use these features to create distinct sections in the store. Consider how clustering and data visualization can guide organizational decisions. This is a key aspect of data science — not only extracting insights from data but also applying them in a practical, business-oriented context.

Based on the results, I would choose the features of flavanoids and total phenols since they have on average the most positive contributions, making them more impactful for wine distinction. I would use these features to create distinct sections in the store as follows:

1. Flavanoid content: Create sections ranging from low to high flavanoid content, which can help customers choose wines based on their flavanoid.
2. Total phenols: Organize wines from low to high total phenol content.

This organization would allow customers to easily find wines that match their preferences for taste profiles. Additionally, it provides an opportunity to educate customers about these wine characteristics and their significance.

Clustering and data visualization have guided this organizational decision by revealing which features are most influential in differentiating wines. The K-means clustering results and PCA analysis have shown that flavanoids and total phenols are key factors in distinguishing between different types of wines. By applying these insights to the store layout, we're translating from the data visualization findings to a practical, business-oriented context that can enhance the customer experience.

Question 6 (0.5 points)

Do you see any outliers in the K-means result? What do they signify? What would you do about these in a similar context to wine selling?

In the first scatterplot created before applying the PCA, there seemed to be more mixture of the pruple and yellow cluster, and it is possible to identify some outliers in the data. However, after applying the PCA, the data seems to be more separated and the outliers are less visible. Some of this outliers could be due to the fact that the data is not normalized, so some features have more weight than others, this causes the data to be skewed and the outliers to be more visible. What i would do about these outliers is normalize the data so that all features have the same weight and the data is not skewed in order to be able to sell wines with more accuracy.

Question 7 (0 points)

Did you use an LLM like ChatGPT or Claude to assist in answering this problem set?

Write "No" if you did not. Write "Yes" and paste a link to the transcript (e.g. https://chat.openai.com/share/5c14a304-1b7f-4fb9-b400-21e65ad545bb) if you did.

I didnt use Chatgpt or any other LLM to assist in answering this problem set.

Question 8 (0 points)

Please use the link below to provide feedback on how well the assignment aligned with the concepts covered in class. Your input will help us improve and refine future assignments.

Form Link - https://forms.gle/LtPwzFayDMUyBcay6

Did you fill out the feedback form?

yes

Submission

Follow the instructions below, then upload the ZIP file to bCourses.

Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

In [51]:

```
# Save your notebook first, then run this cell to export your submission.
grader.export(pdf=False)
```

Your submission has been exported. Click here to download the zip file.