

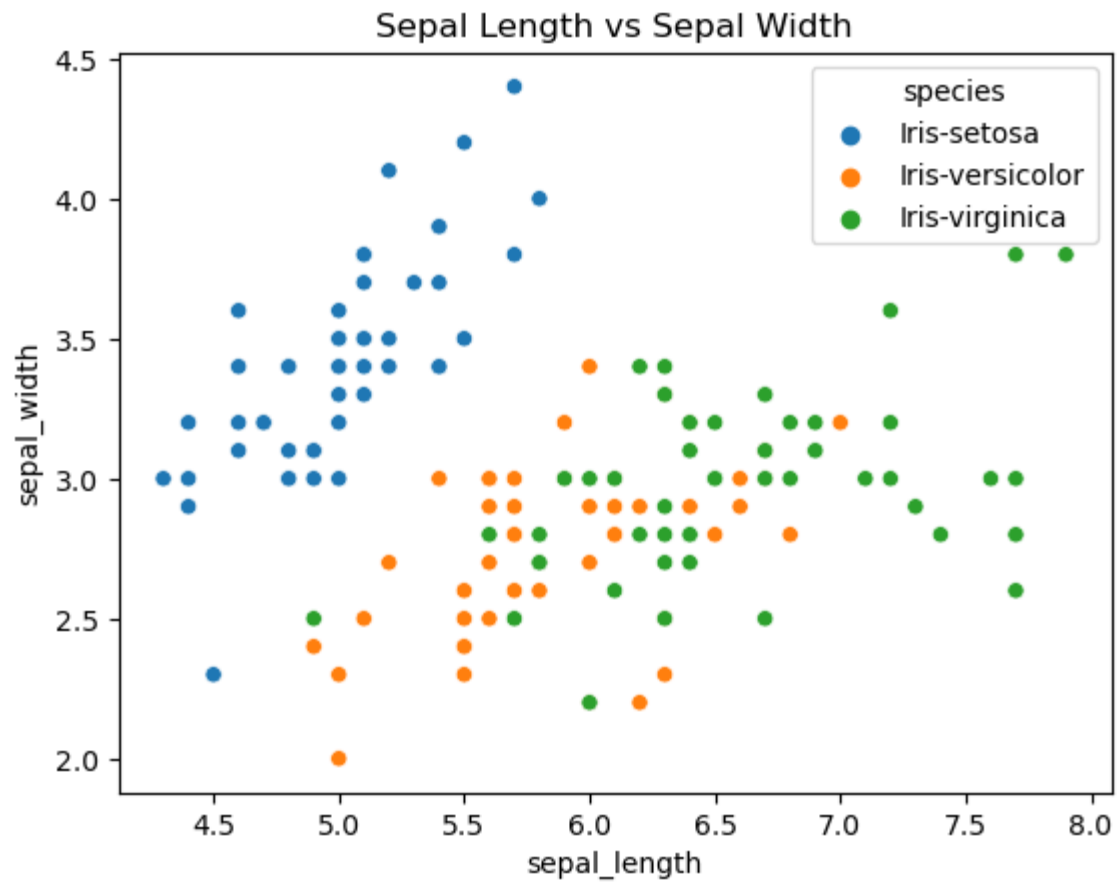
```
In [1]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset into a DataFrame
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
df = pd.read_csv(url, header=None, names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])
```

Scatter Plot:

Plotting pairs of features against each other to see their relationships. This is particularly useful for comparing features against each other.

```
In [2]: sns.scatterplot(x='sepal_length', y='sepal_width', data=df, hue='species')
plt.title('Sepal Length vs Sepal Width')
plt.show()
```



It represents the relationship between the sepal length and sepal width of three different species of Iris flowers: Iris-setosa, Iris-versicolor, and Iris-virginica. The data points are color-coded according to the species.

The x-axis represents the sepal length, ranging from 4.5 to 8.0.

The y-axis represents the sepal width, ranging from 2.0 to 4.5.

Blue dots represent Iris-setosa, orange dots represent Iris-versicolor, and green dots represent Iris-virginica.

we can observe:

1)"Iris-setosa (blue dots) tend to have a higher sepal width and lower sepal length."

2)"Iris-versicolor (orange dots) and Iris-virginica (green dots) show a trend of increasing sepal length with decreasing sepal width."

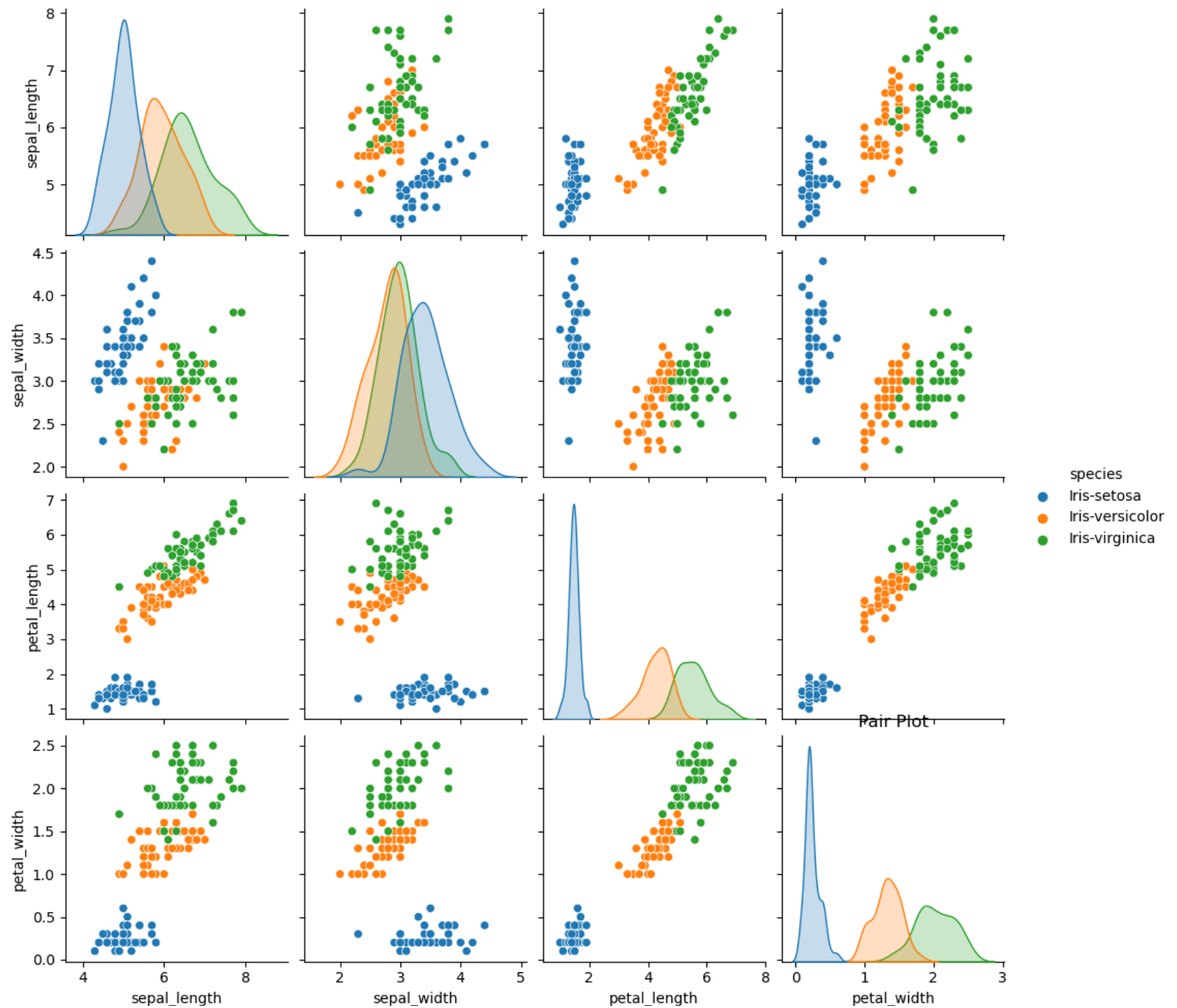
3)"Iris-setosa is quite distinct from the other two species in terms of sepal dimensions."

Pair Plot:

A grid of scatterplots for all pairs of features in the dataset. This provides a quick overview of the relationships between all pairs of features.

```
In [3]: # Pair Plot
sns.pairplot(df, hue='species')
plt.title('Pair Plot')
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)
```



Pair Plot visualization consisting of scatter plots and histograms representing data from three different species of iris flowers: setosa, versicolor, and virginica.

the blue color represents setosa, green represents versicolor, and orange represents virginica.

The scatter plots reveal relationships between pairs of features, and the histograms display frequency distributions, each color-coded to represent one of the three iris flower species.

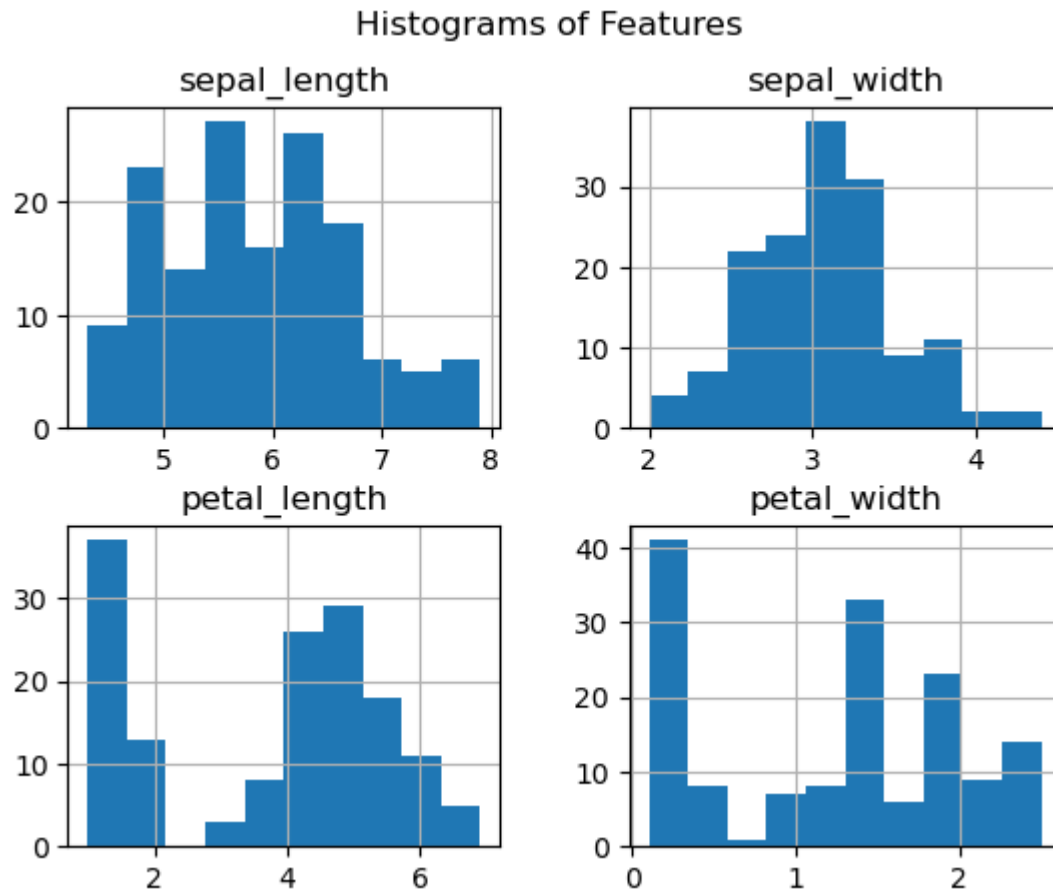
useful for the decision making in:

- 1)Species Identification: The scatter plots and histograms can be used to identify and classify the species of iris flowers based on sepal length, sepal width, petal length, and petal width.
- 2)Feature Distribution: The histograms on the diagonal show the distribution of each feature for different species, which can aid in understanding the variability within each species.
- 3)Correlation Analysis: Scatter plots below the diagonal help in visualizing relationships or correlations between pairs of features. This is crucial for selecting features that are significant predictors for classification tasks.
- 4)Classification Models: The distinct clusters formed by different colors (species) in scatter plots indicate that machine learning models can effectively be trained to classify iris species based on these features.

Histogram:

Plotting histograms of each feature individually to see the distribution of values. This can help in understanding the range and spread of each feature.

```
In [4]: # Histogram  
df.hist()  
plt.suptitle('Histograms of Features')  
plt.show()
```



four histograms representing features of flowers: sepal length, sepal width, petal length, and petal width.

each histogram is labeled with its corresponding feature name at the bottom

The title "Histograms of Features" is displayed at the top center position.

1)Data Analysis: The histograms provide a visual representation of the distribution of sepal and petal lengths and widths, which can be analyzed to understand patterns and variations.

2)Feature Selection: By observing the spread and overlap of different features, decisions regarding which features are significant for classification or other analyses can be made.

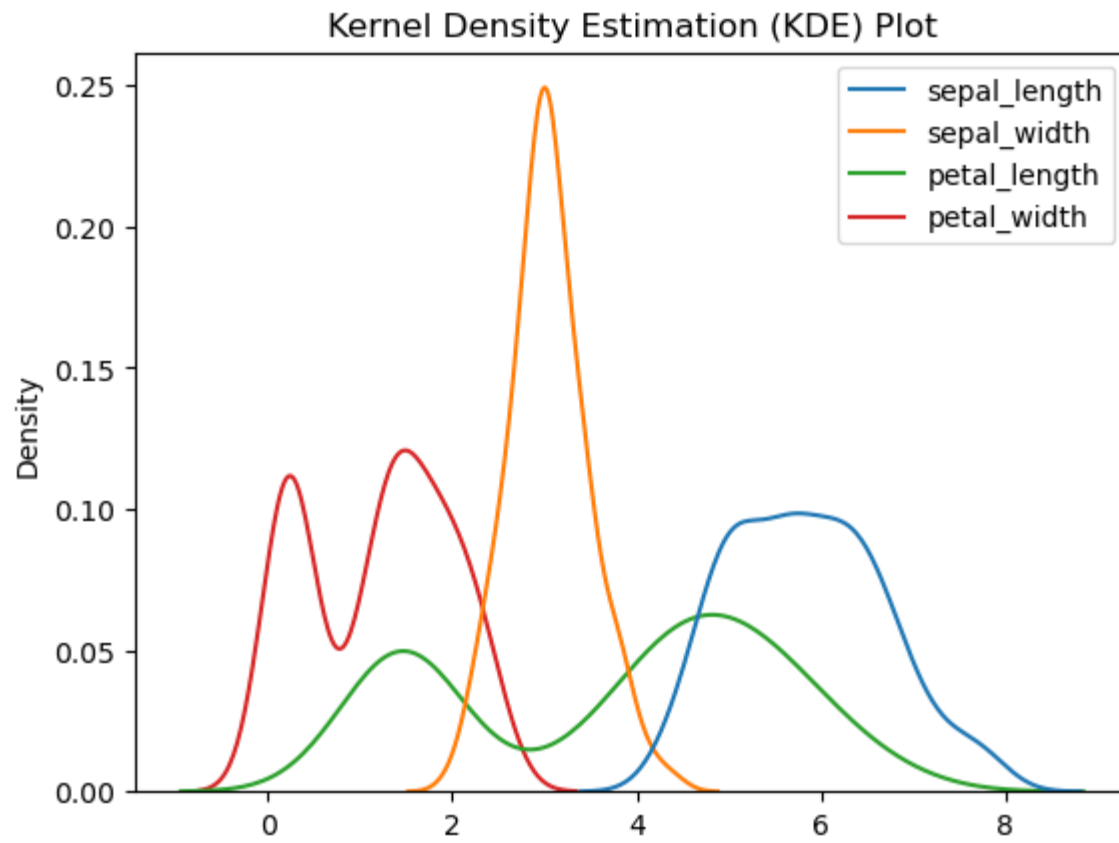
3)Model Training: The data distributions can inform the selection of appropriate machine learning models or algorithms to use for tasks like classification or clustering.

4)Insights Extraction: Analyzing these histograms can provide insights into the characteristics of different flower species, aiding in their identification.

Kernel Density Estimation (KDE) Plot:

Similar to histograms but shows a smooth estimation of the probability density function of a continuous variable.

```
In [5]: # Kernel Density Estimation (KDE) Plot
sns.kdeplot(data=df)
plt.title('Kernel Density Estimation (KDE) Plot')
plt.show()
```



multi-colored line graph representing Kernel Density Estimation (KDE) plots of four different variables related to flowers' sepals and petals.

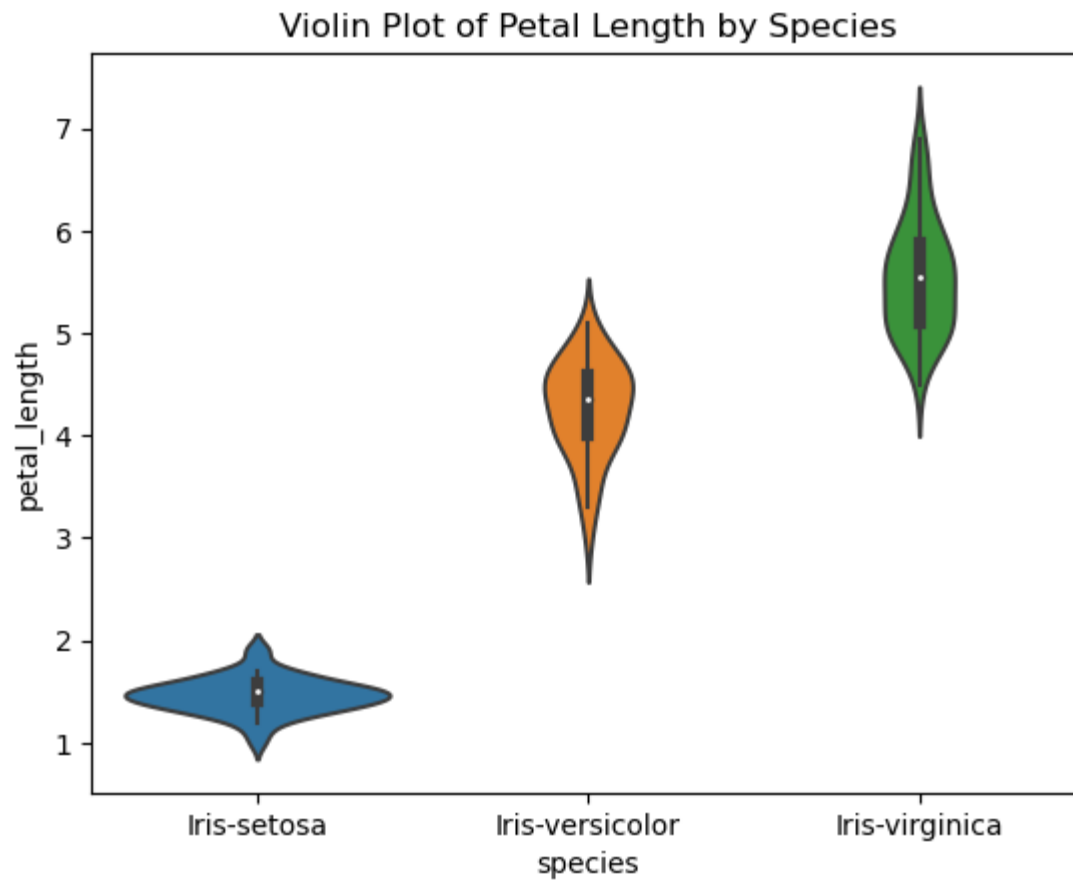
each KDE plot is labeled with its corresponding feature name at the bottom. The title "Kernel Density Estimation (KDE) Plot" is displayed at the top center position.

- 1)Data Analysis: The KDE plots provide a visual representation of the distribution of sepal and petal lengths and widths, which can be analyzed to understand patterns and variations.
- 2)Feature Selection: By observing the spread and overlap of different features, decisions regarding which features are significant for classification or other analyses can be made.
- 3)Model Training: The data distributions can inform the selection of appropriate machine learning models or algorithms to use for tasks like classification or clustering.
- 4)Insights Extraction: Analyzing these KDE plots can provide insights into the characteristics of different flower species, aiding in their identification.

Violin Plot:

Combines a box plot with a KDE plot. It shows the distribution of the data across different categories.

```
In [6]: # Violin Plot
sns.violinplot(x='species', y='petal_length', data=df)
plt.title('Violin Plot of Petal Length by Species')
plt.show()
```



“Violin Plot of Petal Length by Species”. It represents the distribution of petal lengths for three different species of Iris flowers: Iris-setosa, Iris-versicolor, and Iris-virginica.

each violin plot is color-coded; blue for Iris-setosa, orange for Iris-versicolor, and green for Iris-virginica.

The y-axis is labeled “petal length” and ranges from 0 to 7.

The x-axis labels indicate the species names.

Each violin plot shows the distribution of petal lengths with a white dot indicating the median value.

1)Species Identification: The plot helps in distinguishing between the three species of Iris flowers based on petal length.

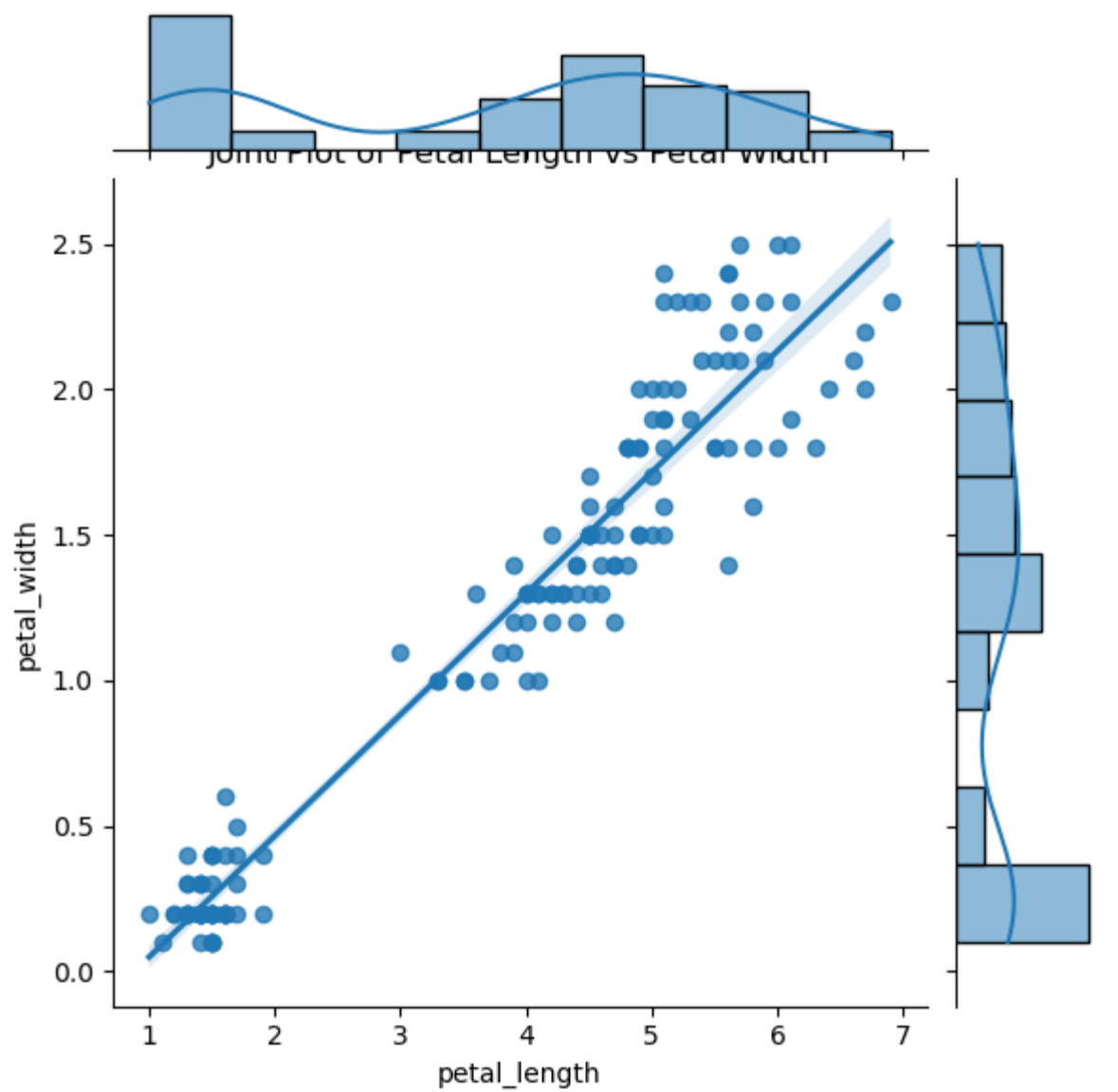
2)Data Visualization: It provides a visual representation of the distribution and density of petal lengths for each species, aiding in quick data interpretation.

3)Statistical Analysis: The plot can be used to observe the median, spread, and distribution skewness of petal lengths, assisting in statistical analysis.

Joint Plot:

Plots a combination of scatter plots and histograms, including a regression line (if desired), to visualize the relationship between two variables and their distributions.

```
In [7]: # Joint Plot
sns.jointplot(x='petal_length', y='petal_width', data=df, kind='reg')
plt.title('Joint Plot of Petal Length vs Petal Width')
plt.show()
```



“Joint Plot of Petal Length vs Petal Width”. It represents the relationship between petal length and width.

blue dots represent individual data points showing various combinations of petal lengths and widths. A blue line indicates a positive linear relationship between petal length and width; as one increases, so does the other.

1)Data Analysis: It provides a visual representation of the relationship between petal length and width, which can be crucial in analyzing data related to flowers.

2)Correlation Identification: The scatter plot helps in identifying if there is a correlation between the two variables. In this case, there appears to be a positive correlation.

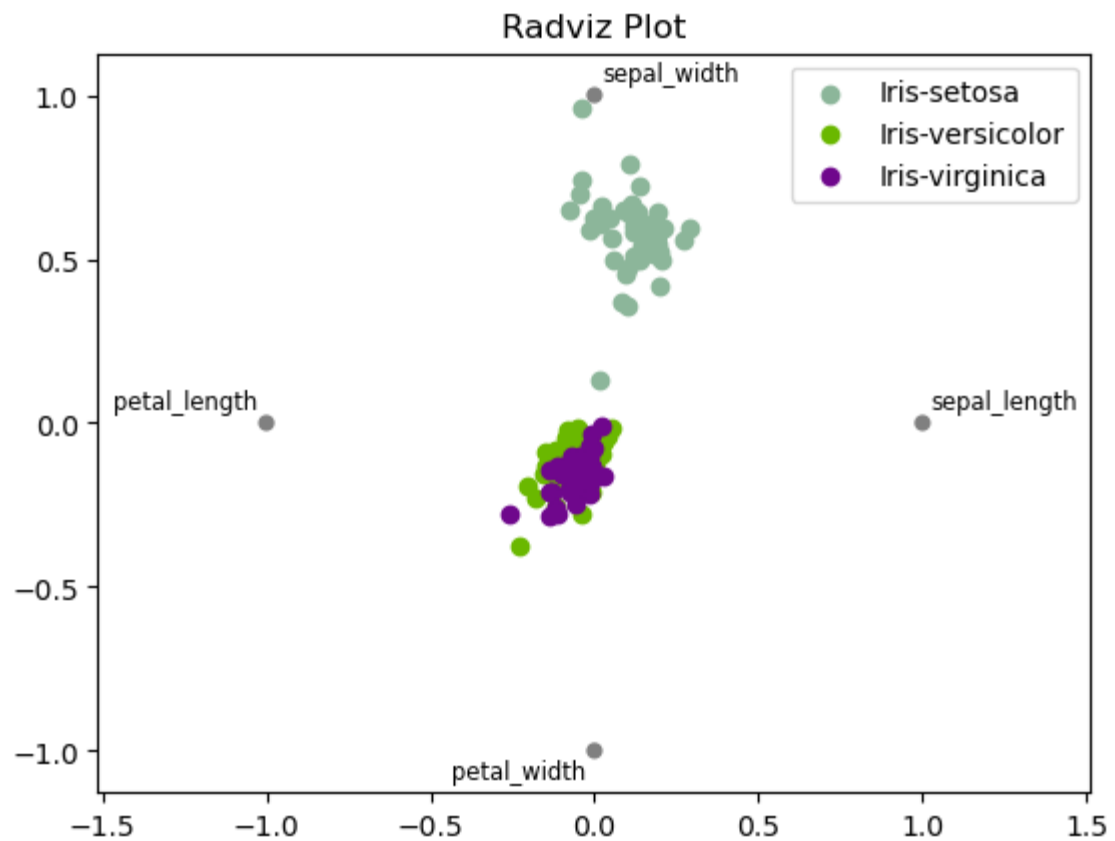
3)Trend Analysis: The trend line aids in predicting future values and understanding the pattern or trend of the data.

4)Data Distribution: The histograms on top and right side give insights into the distribution of petal lengths and widths respectively.

Radviz Plot:

A technique for visualizing multivariate data. It places each feature as a point on a 2D plane and then simulates a spring-like force between each point and the center point based on the feature values.

```
In [8]: # Radviz Plot
from pandas.plotting import radviz
radviz(df, 'species')
plt.title('Radviz Plot')
plt.show()
```



Radviz Plot used for multivariate data visualization.

three clusters of data points are visible, each representing a different species of Iris flowers: Iris-setosa (blue), Iris-versicolor (green), and Iris-virginica (purple).

Four anchors labeled as "sepal_width," "sepal_length," "petal_width," and "petal_length" represent the features measured from each flower sample. Each data point's position is determined by the values of these four features.

1)Species Identification: It helps in distinguishing between different species of Iris flowers based on sepal and petal dimensions.

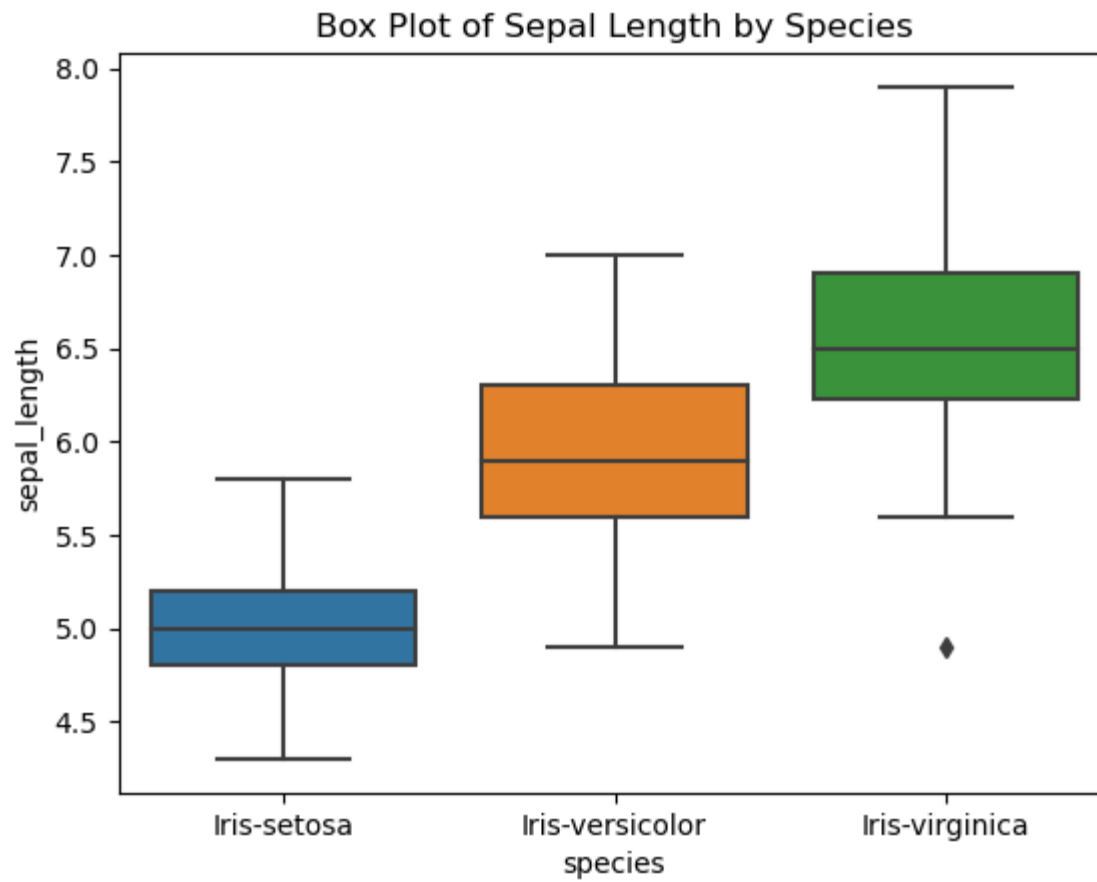
2)Feature Importance: Identifying which features (sepal width, sepal length, petal width, petal length) are significant in classifying the species.

3)Data Visualization: Offers a visual representation for complex data to aid in understanding patterns and correlations.

Box Plot:

Shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable.

```
In [10]: # Box Plot
sns.boxplot(x='species', y='sepal_length', data=df)
plt.title('Box Plot of Sepal Length by Species')
plt.show()
```



“Box Plot of Sepal Length by Species”. It represents the distribution of sepal lengths for three different species of Iris flowers: Iris-setosa, Iris-versicolor, and Iris-virginica.

each box plot displays the distribution of sepal lengths for each species with a line indicating the median value. There's an outlier indicated with a black dot near the 8.0 mark on the y-axis for Iris-virginica.

1)Species Identification: The box plot helps in distinguishing between different species of Iris flowers based on sepal length.

2)Data Analysis: It provides a visual representation of the distribution of sepal lengths for each species, aiding in quick data interpretation.

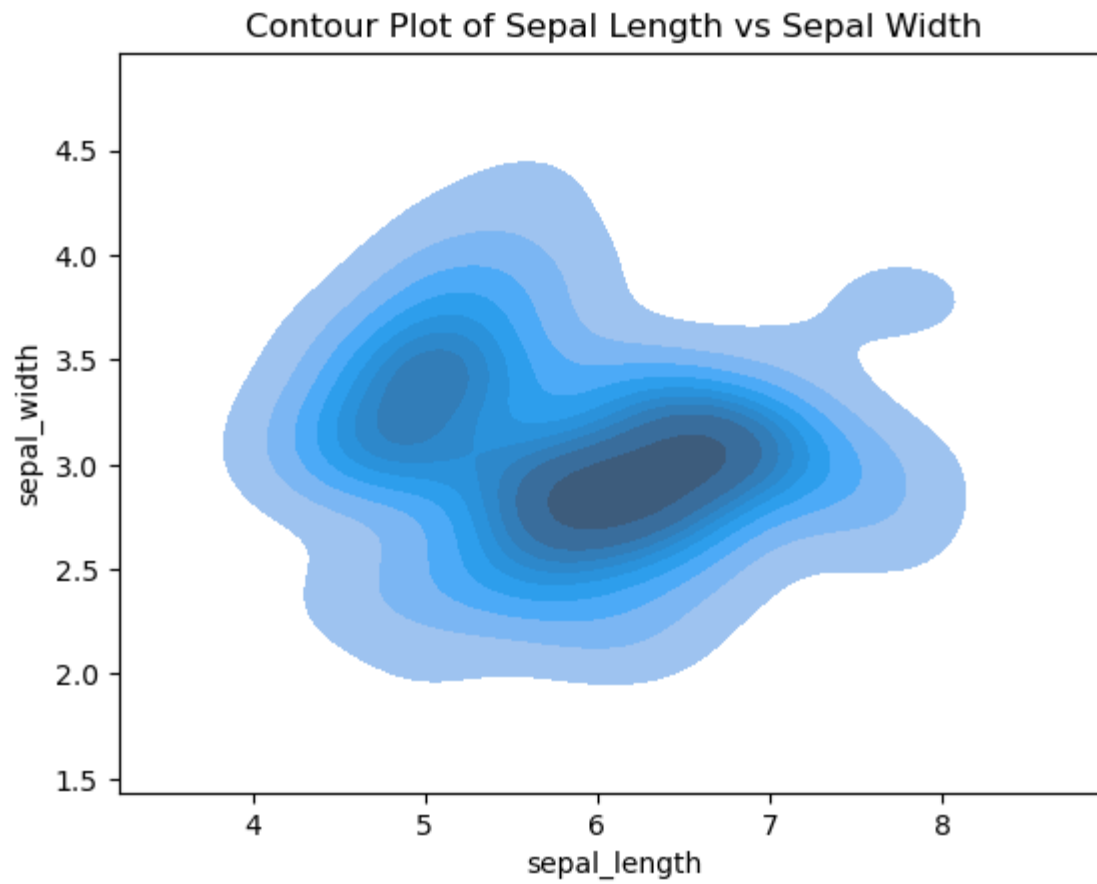
3)Statistical Analysis: The plot can be used to observe the median, quartiles, and potential outliers in the data. This information is essential for statistical analysis and predictive modeling.

4)Feature Importance: By observing the spread and overlap of sepal lengths among different Iris species, decisions regarding which features are significant for classification or other analyses can be made.

Contour Plot:

Useful for visualizing three-dimensional data in two dimensions. For example, you can use it to plot the relationship between sepal length and width, with a contour indicating the density of data points.

```
In [11]: # Contour Plot
sns.kdeplot(x='sepal_length', y='sepal_width', data=df, fill=True)
plt.title('Contour Plot of Sepal Length vs Sepal Width')
plt.show()
```



The image is a contour plot titled "Contour Plot of Sepal Length vs Sepal Width".

The x-axis represents "sepal_length" ranging from approximately 4 to 8, and the y-axis represents "sepal_width" ranging from approximately 1.5 to 4.5.

The contours, represented by different shades of blue, indicate the density of data points at those particular coordinates, with darker shades indicating higher densities.

plot provides a visual representation of the distribution of sepal length and width, which can aid in quick insights and understanding of the data.

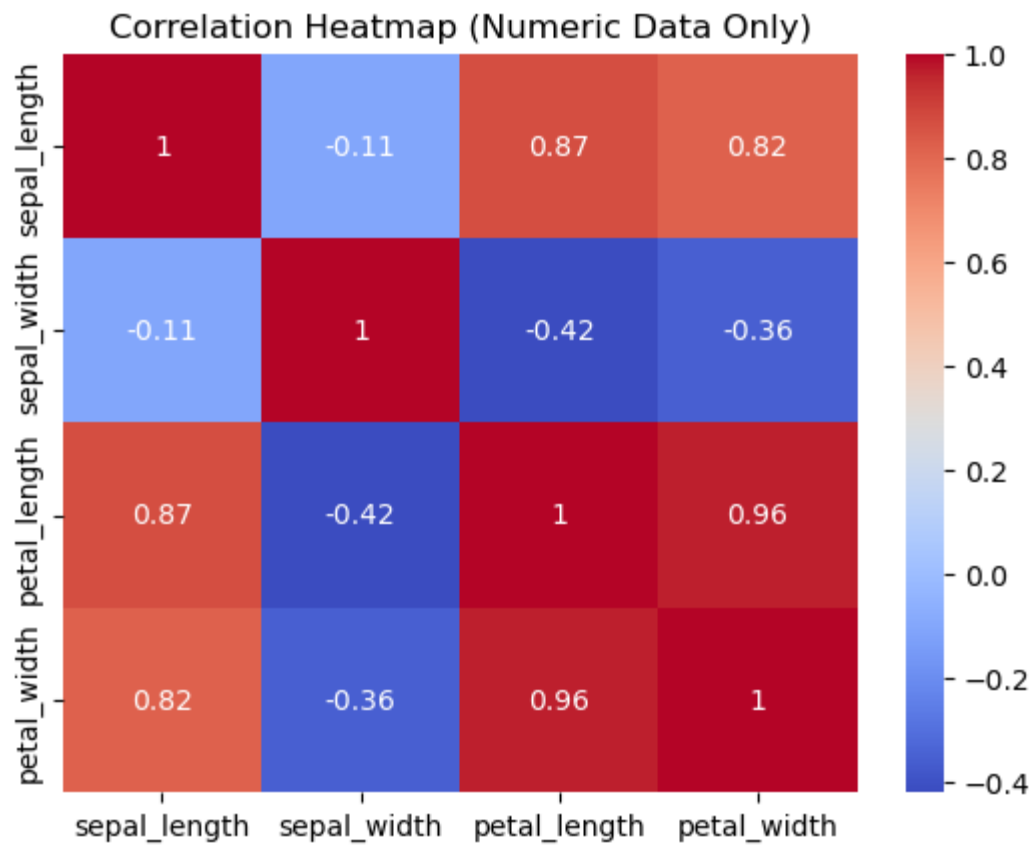
contour plot helps in identifying patterns and correlations between sepal length and width. For instance, the darkest area centered around a sepal length of about 6 and a sepal width of about 3 indicates the most common sepal dimensions.

Heatmap:

A graphical representation of data where values are depicted by colors. In the context of the Iris dataset, you can create a correlation heatmap to visualize the correlation between different features. Note:- Drop the non-numeric column and then plot the heatmap. ('species' is the non-numeric column in this dataset)

```
In [12]: # Drop the non-numeric 'species' column
numeric_df = df.drop('species', axis=1)

In [13]: # Compute correlations and plot heatmap
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap (Numeric Data Only)')
plt.show()
```



The image is a correlation heatmap visualizing the relationships between four different numerical data types: sepal length, sepal width, petal length, and petal width.

Each cell in the heatmap represents the correlation coefficient between two variables.

The color gradient from red to blue represents correlation values from 1 to -0.4 respectively; red indicates strong positive correlation while blue indicates negative correlation.

This heatmap shows correlations between different measurements of flowers, which can be used to understand relationships between these variables. For example, the cell at the intersection of "petal_length" and "petal_width" has a value of 0.96 indicating a strong positive correlation.

Positive correlations (closer to 1) indicate a direct relationship; as one variable increases, the other also increases.

Negative correlations (closer to -1) indicate an inverse relationship; as one variable increases, the other decreases.

```
In [2]: # noise removal by binning
```

```
In [ ]:
```

```
In [ ]:
```