

dv-with-matplotlib-seaborn-3

September 17, 2024

1 Data visualization with matplotlib and seaborn using the Iris Dataset

->Matplotlib:Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Common Plot Types in Matplotlib:

Line plots for continuous data.

Scatter plots for visualizing the relationship between two numerical variables.

Histograms for understanding the distribution of data.

->Seaborn:Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative

statistical graphics.

Plot Types in Seaborn:

Pair plots to show pairwise relationships in a dataset.

Box plots and violin plots for visualizing the distribution of data.

Heatmaps for visualizing the correlation between variables.

1.General Statistics Plot (Matplotlib or Seaborn):

```
[4]: #importing seaborn and matplotlib libraries
```

```
import seaborn as sns
import matplotlib.pyplot as plt

#Load the iris dataset

iris=sns.load_dataset('iris')
print(iris)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa

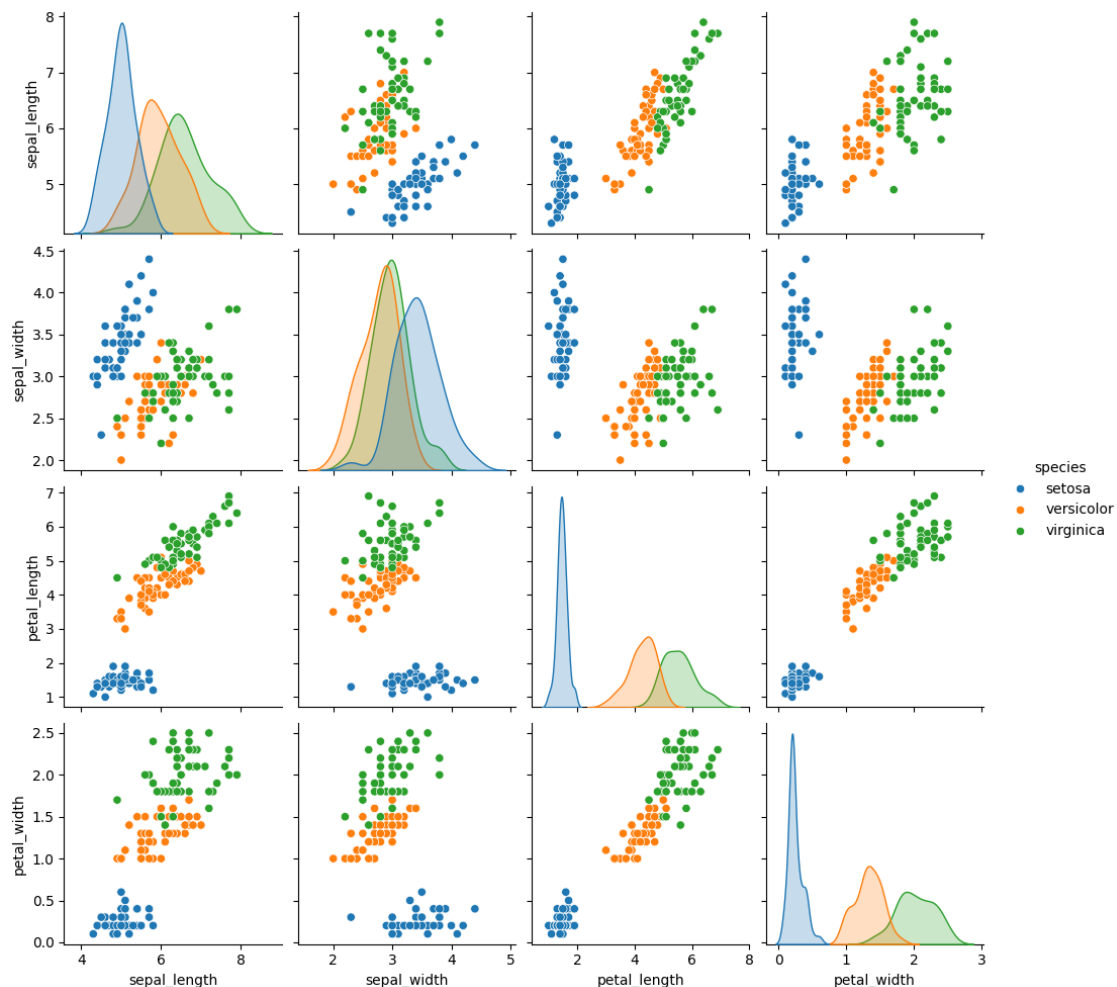
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

[150 rows x 5 columns]

PAIRPLOT: A pair plot, also known as a scatterplot matrix, is a matrix of graphs that enables the visualization of the relationship between each pair of variables in a dataset.

```
seaborn.pairplot(data, *, hue=None, hue_order=None, palette=None, vars=None, x_vars=None,
y_vars=None, kind='scatter', diag_kind='auto', markers=None, height=2.5, aspect=1, corner=False, dropna=False, plot_kws=None, diag_kws=None, grid_kws=None, size=None)
```

```
[7]: sns.pairplot(iris, hue='species')
plt.show()
```

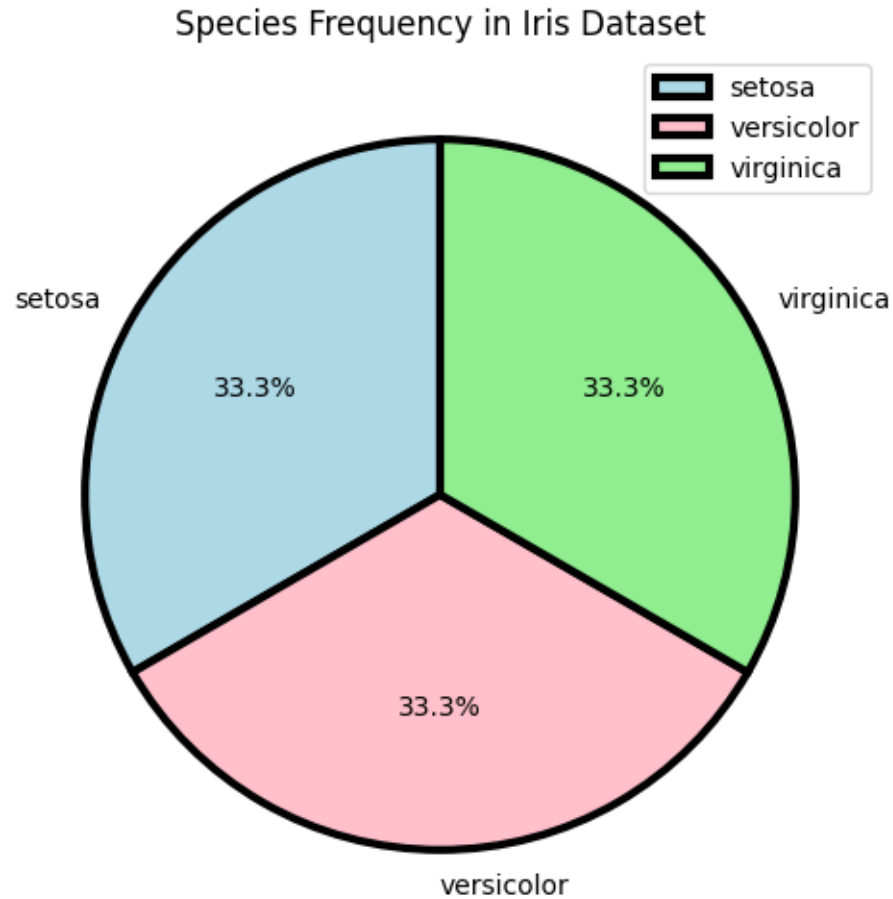


2. Pie Plot for Species Frequency:

PIE PLOT: A Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data.

`matplotlib.pyplot.pie(x, explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=None, radius=None, counterclock=True, wedgeprops=None, textprops=None, center=(0, 0), frame=False, rotatelabels=False, *, data=None)`[\[source\]](#)

```
[20]: species_counts = iris['species'].value_counts()
plt.figure(figsize=(6,6))
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%%',
    ↪startangle=90, colors=['lightblue', 'pink', 'lightgreen'], wedgeprops=
    ↪{"edgecolor": "black",
        'linewidth': 3, 'antialiased': True})
plt.title('Species Frequency in Iris Dataset')
plt.legend()
plt.show()
```

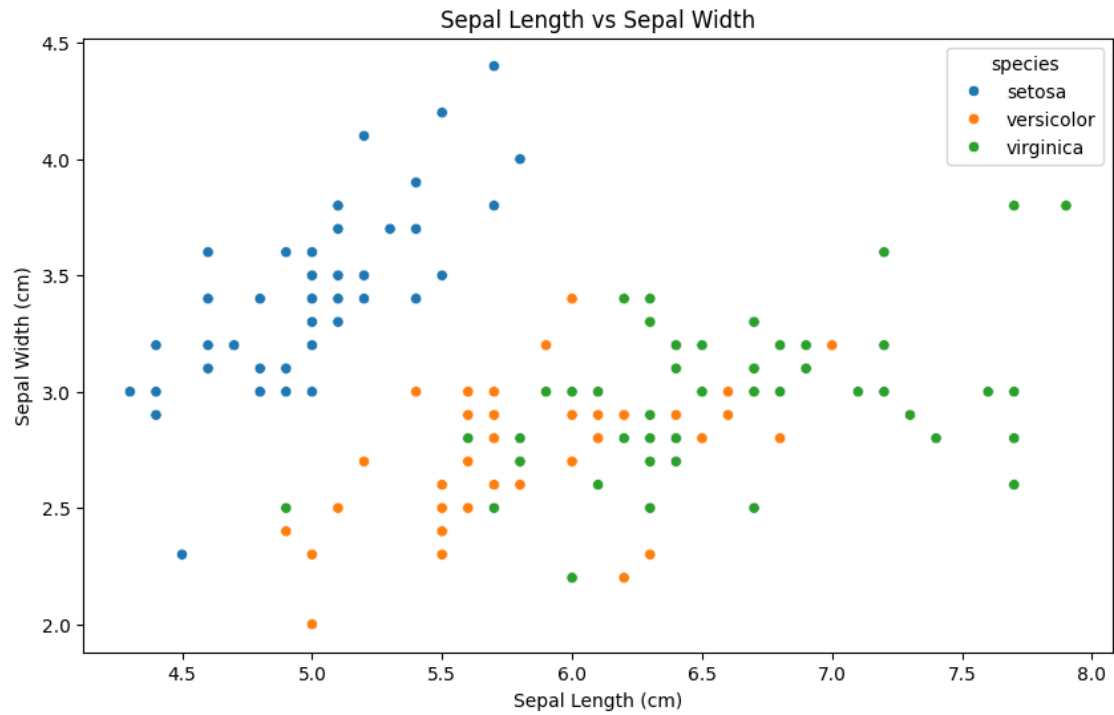


3. Relationship Between Sepal Length and Width:

SCATTER PLOT: A scatter plot is a diagram where each value in the data set is represented by a dot.

```
seaborn.scatterplot(data=None, *, x=None, y=None, hue=None, size=None, style=None,
palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None,
size_norm=None, markers=True, style_order=None, legend='auto', ax=None, **kwargs)
```

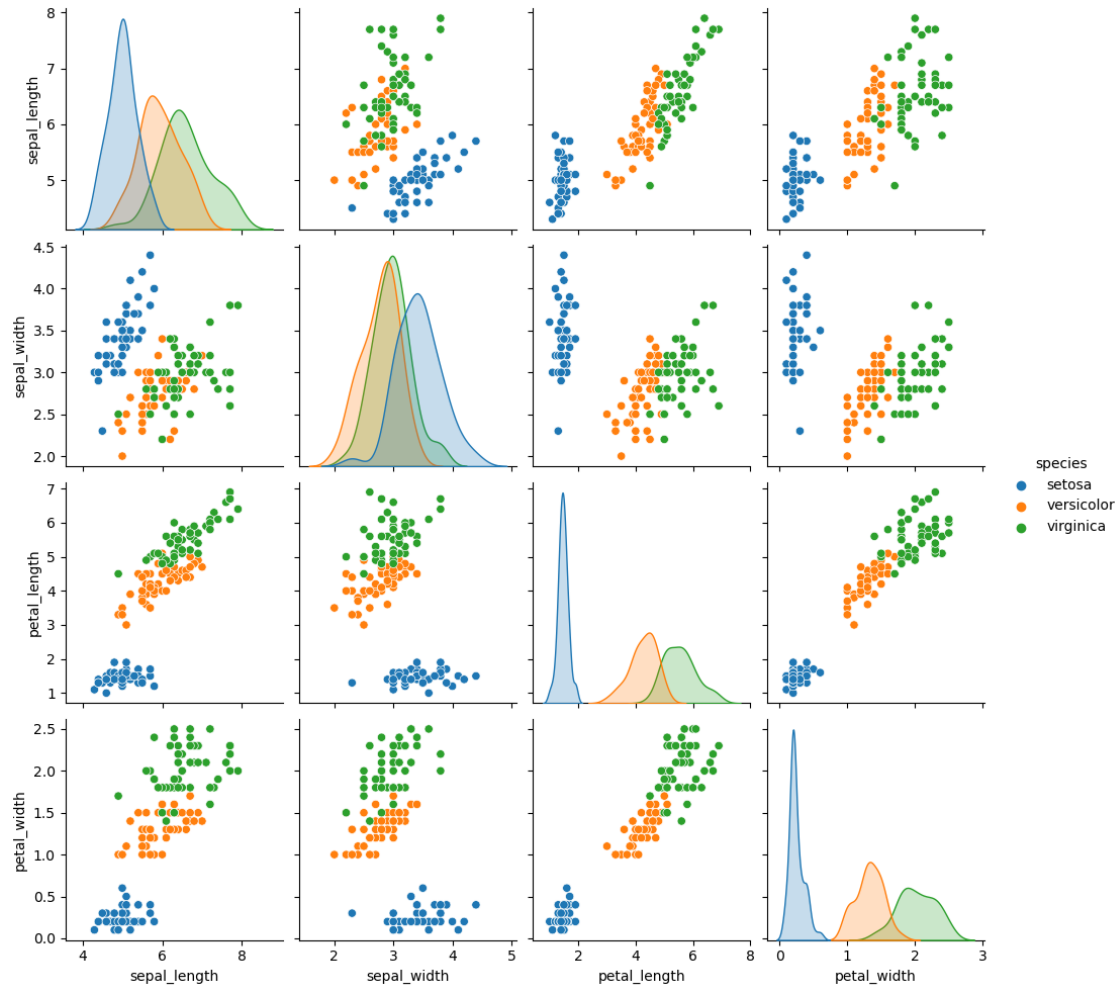
```
[25]: plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=iris)
plt.title('Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.show()
```



4. Distribution of Sepal and Petal Features:

```
[25]: sns.pairplot(iris, hue='species', height=2.5)  
plt.show()
```

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

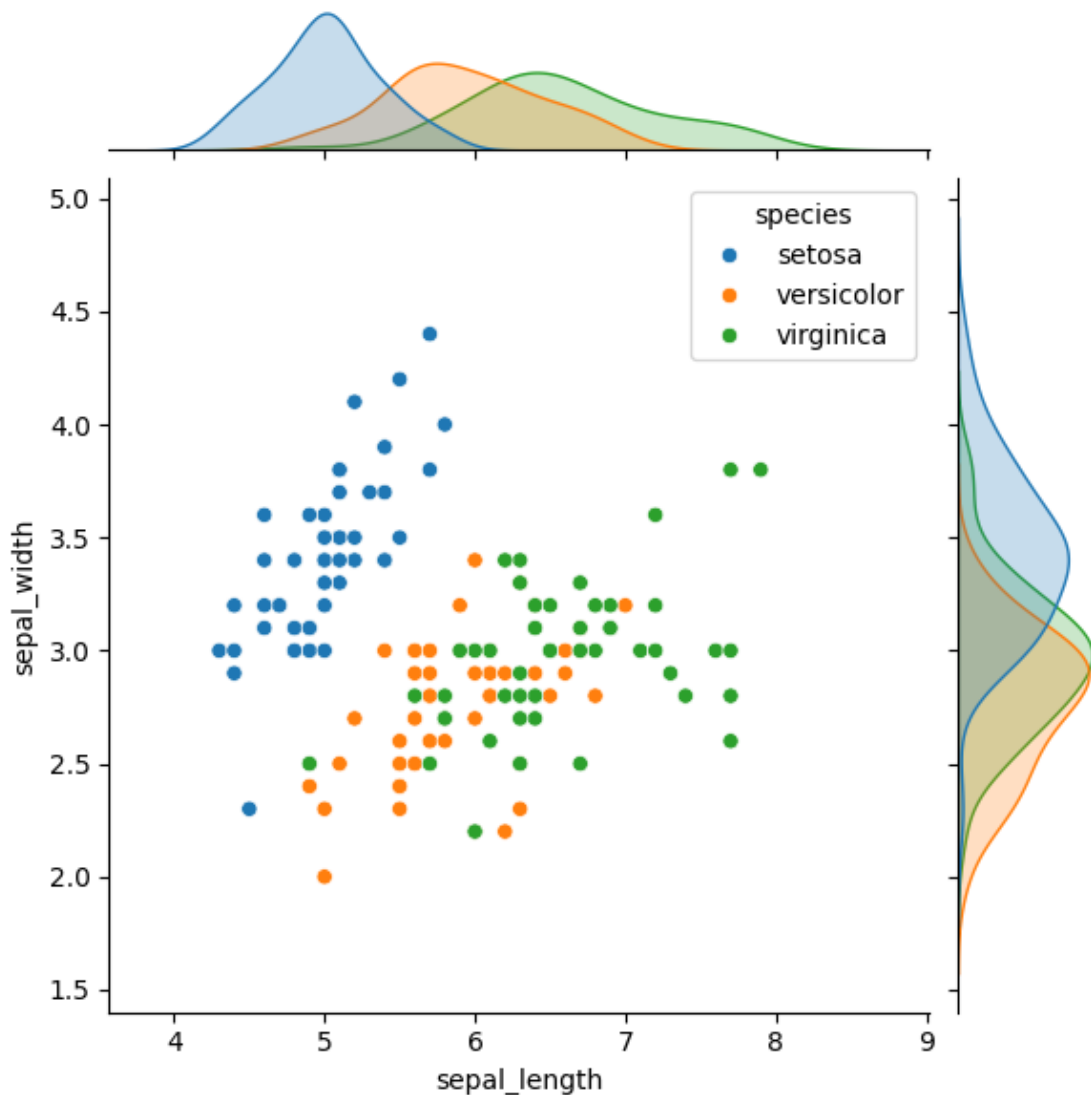


5. Jointplot of Sepal Length vs Sepal Width:

JOINT PLOT: The joint plot is a way of understanding the relationship between two variables and the distribution of individuals of each variable. The joint plot mainly consists of three separate plots in which, one of them was the middle figure that is used to see the relationship between x and y.

```
seaborn.jointplot(data=None, *, x=None, y=None, hue=None, kind='scatter', height=6, ratio=5,
space=0.2, dropna=False, xlim=None, ylim=None, color=None, palette=None, hue_order=None,
hue_norm=None, marginal_ticks=False, joint_kws=None, marginal_kws=None, **kwargs)
```

```
[45]: sns.jointplot(x='sepal_length', y='sepal_width', data=iris,
↳ kind='scatter', hue='species')
plt.show()
```



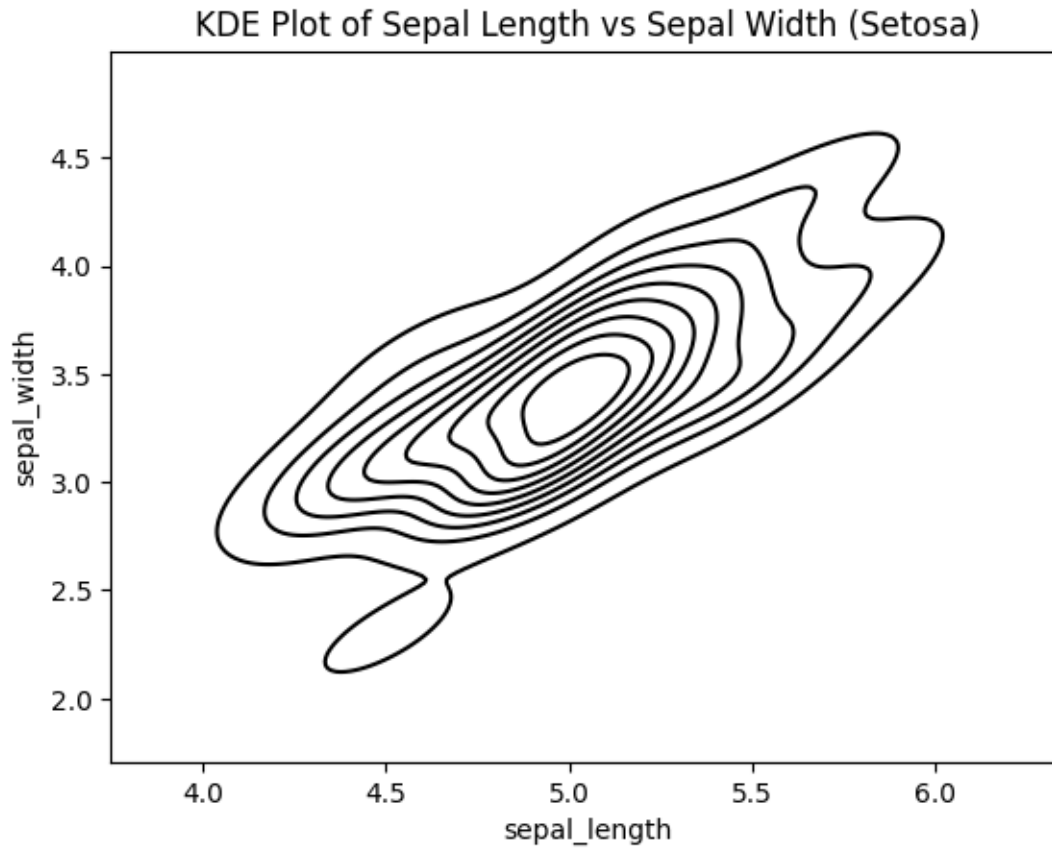
6. KDE Plot for Setosa Species (Sepal Length vs Sepal Width):

A kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset, analogous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions.

```
seaborn.kdeplot(data=None, *, x=None, y=None, hue=None, weights=None, palette=None,
hue_order=None, hue_norm=None, color=None, fill=None, multiple='layer', com-
mon_norm=True, common_grid=False, cumulative=False, bw_method='scott', bw_adjust=1,
warn_singular=True, log_scale=None, levels=10, thresh=0.05, gridsize=200, cut=3, clip=None,
legend=True, cbar=False, cbar_ax=None, cbar_kws=None, ax=None, **kwargs)
```

```
[57]: setosa = iris[iris['species'] == 'setosa']
```

```
sns.kdeplot(x='sepal_length', y='sepal_width',  
            data=setosa,color='black',multiple='fill')  
plt.title('KDE Plot of Sepal Length vs Sepal Width (Setosa)')  
plt.show()
```



7.KDE Plot for Setosa Species (Petal Length vs Petal Width):

```
[47]: sns.kdeplot(x='petal_length', y='petal_width', data=setosa,color='black')  
plt.title('KDE Plot of Petal Length vs Petal Width (Setosa)')  
plt.show()
```