

# data visualization

September 17, 2024

## 1 lab work on data visualization using Matplotlib and Seaborn.

### 1.1 loading the IRIS data set

```
[22]: # importing seaborn and matplotlib
import seaborn as sns
import matplotlib.pyplot as plt
# loading the iris dataset from seaborn
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]

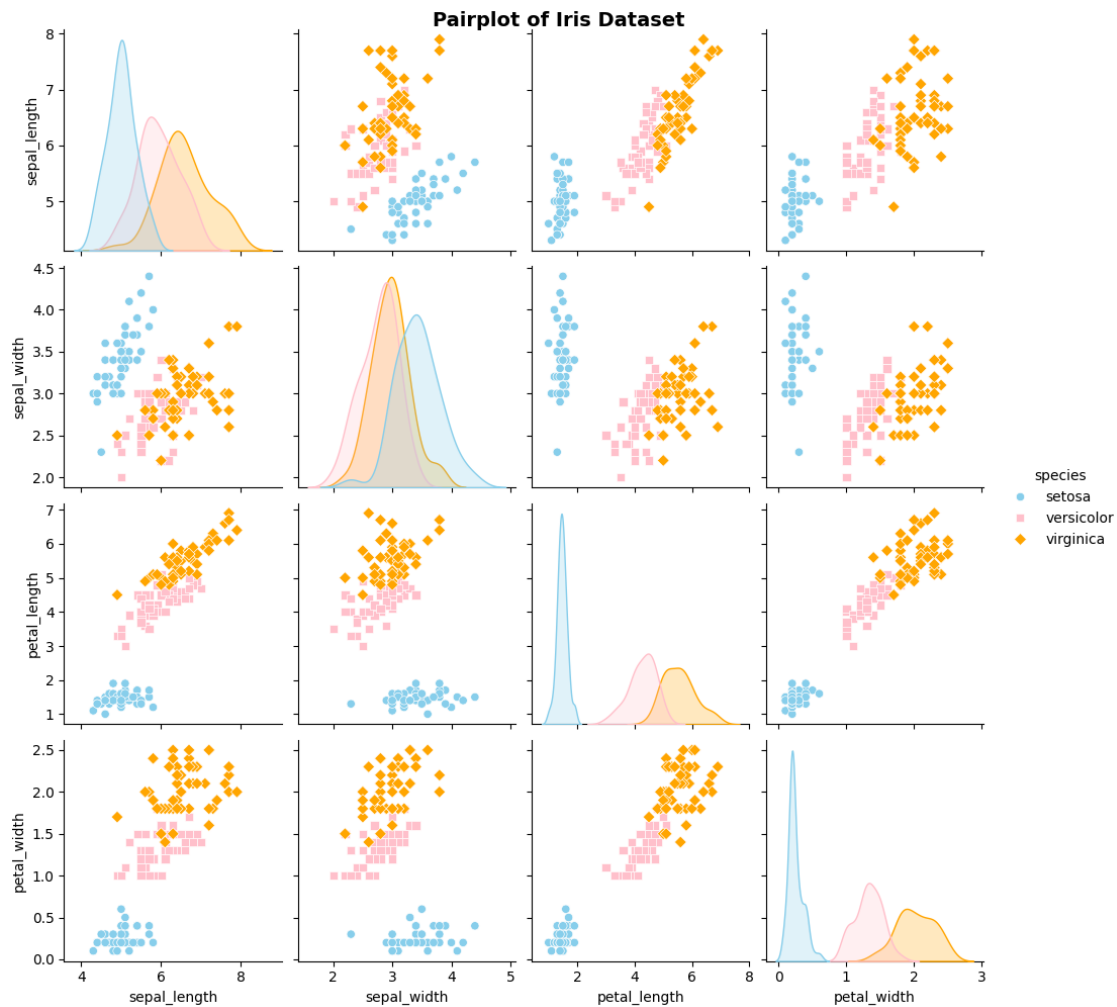
### 1.2 1.General Statistics Plot (Matplotlib or Seaborn):

- Write a Python program to create a plot that gives a general statistical summary of the Iris data. You can use seaborn's pairplot or pandas' describe() for guidance.

```
[23]: # Displaying a general statistical summary
print(iris.describe())
# Creating a pairplot to visualize the relationships and distributions
# hue parameter allows us to differentiate between categories
# palette lets us to customize colors
sns.pairplot(iris, hue='species', markers=["o", "s", "D"],palette=['skyblue', 'pink', 'orange'])
# Showing the plot
```

```
plt.suptitle("Pairplot of Iris Dataset", fontsize=14, fontweight='bold',y=1)
plt.show()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000



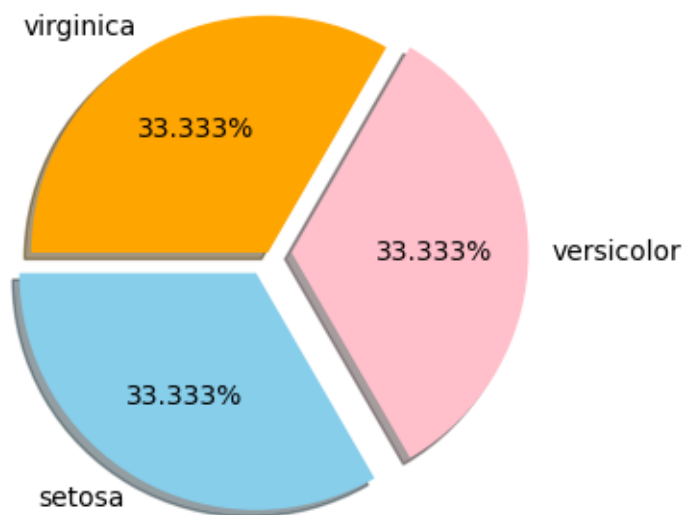
### 1.3 2. Pie Plot for Species Frequency:

- Write a Python program to create a pie chart to display the frequency of the three species (setosa, versicolor, virginica) in the Iris dataset.

```
[24]: # counting the no of sepcies
count=iris['species'].value_counts()
print(count)
print(count.index)
# creating a figure with 5,4 dimentions
plt.figure(figsize=(5,4))
# plotting a pie chart about the frequencies of species
plt.pie(count,labels=count.index,autopct='%1.
↪3f%%',startangle=180,colors=['skyblue', 'pink', 'orange'],shadow=True,
↪explode=(0.1, 0.1, 0))
plt.title('frequency of species in iris dataset',fontsize=16, fontweight='bold')
plt.show()
```

```
species
setosa      50
versicolor  50
virginica   50
Name: count, dtype: int64
Index(['setosa', 'versicolor', 'virginica'], dtype='object', name='species')
```

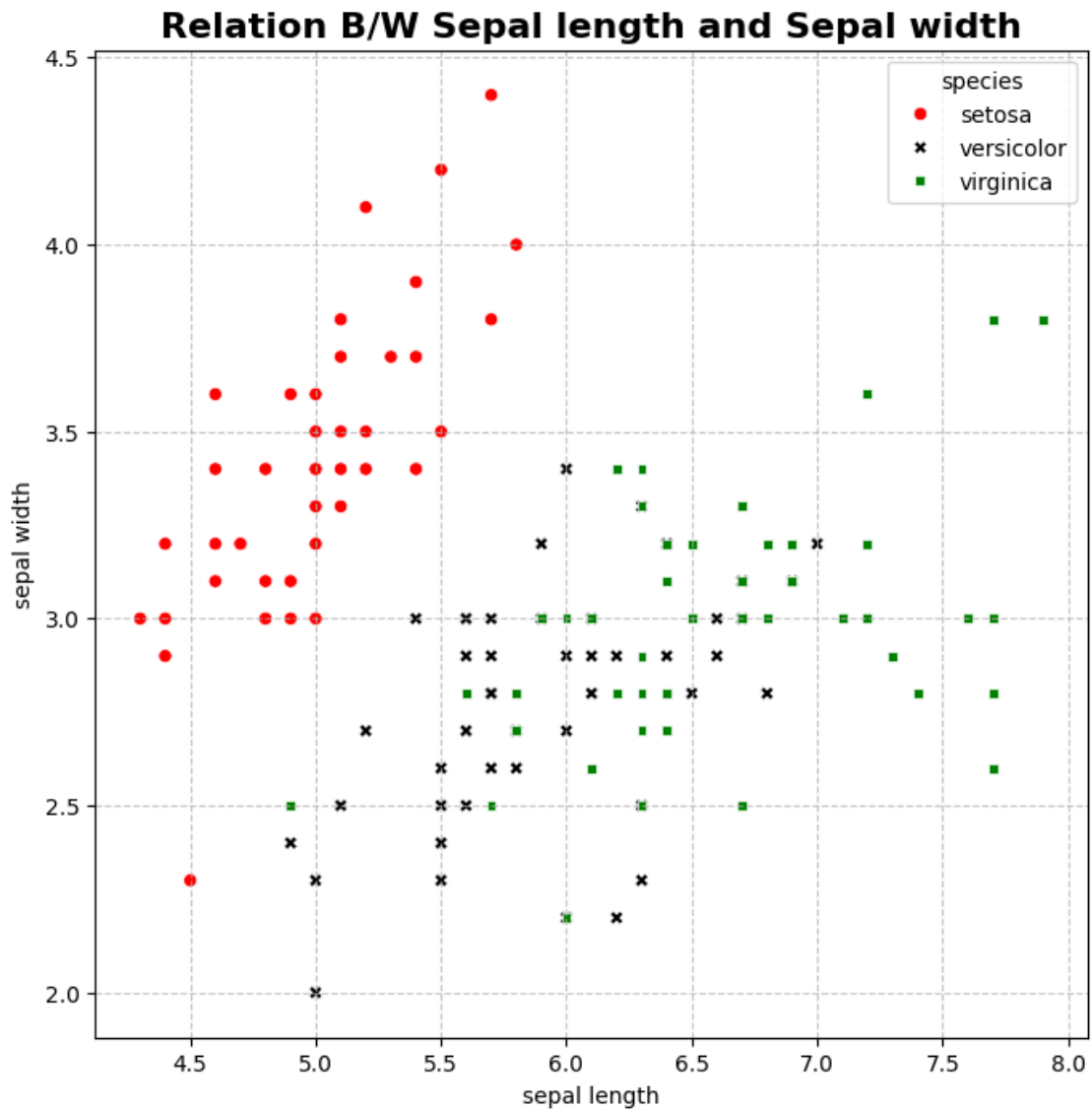
### frequency of species in iris dataset



#### 1.4 3. Relationship Between Sepal Length and Width:

- Write a Python program to create a scatter plot to find the relationship between sepal length and sepal width for the Iris dataset.

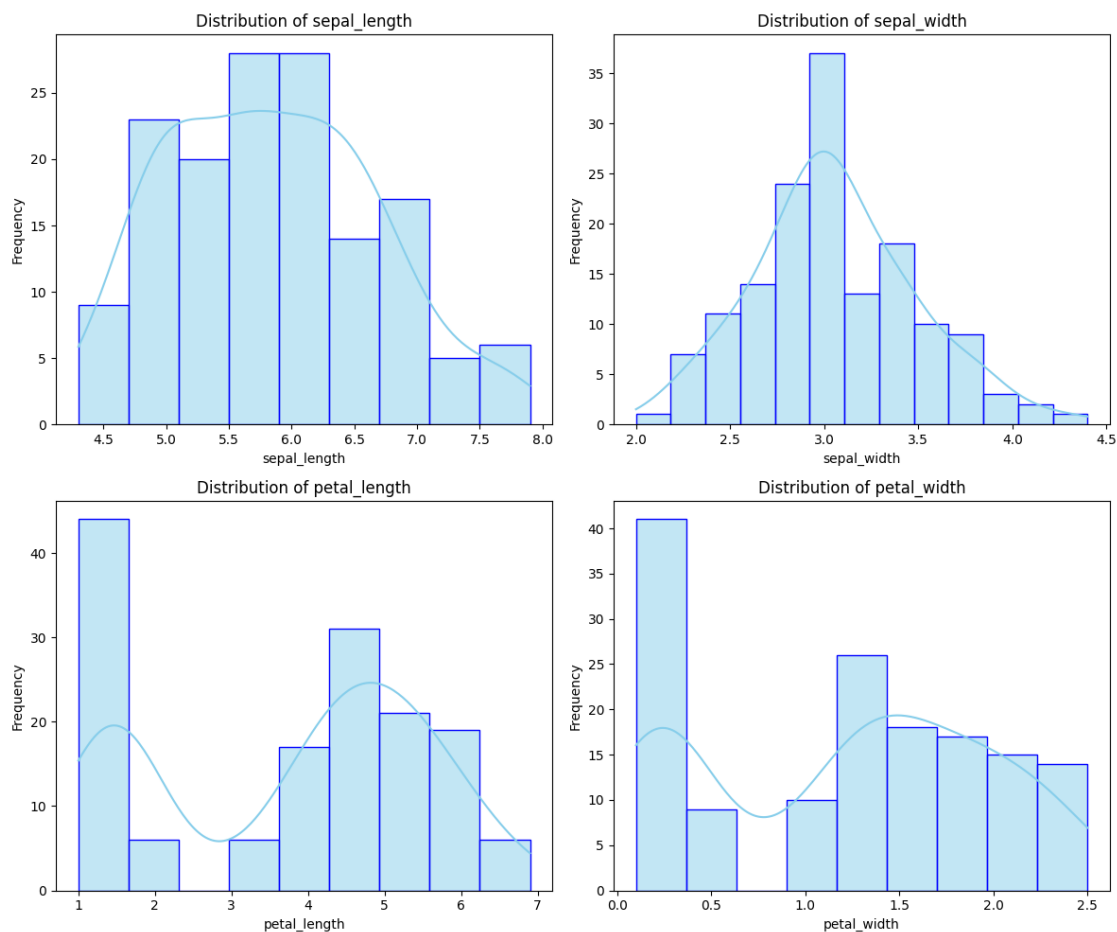
```
[33]: plt.figure(figsize=(8,8))
# plotting a scatter plot to specify the relation b/w sepal length and width
sns.
    ↳ scatterplot(x='sepal_length',y='sepal_width',hue='species',data=iris,palette=['red','black']
plt.title('Relation B/W Sepal length and Sepal width',fontsize=16,
    ↳ fontweight='bold')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
# plotting grid line for more convinience
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



## 1.5 4. Distribution of Sepal and Petal Features:

- Write a Python program to create a plot that shows how the length and width of sepal length, sepal width, petal length, and petal width are distributed.

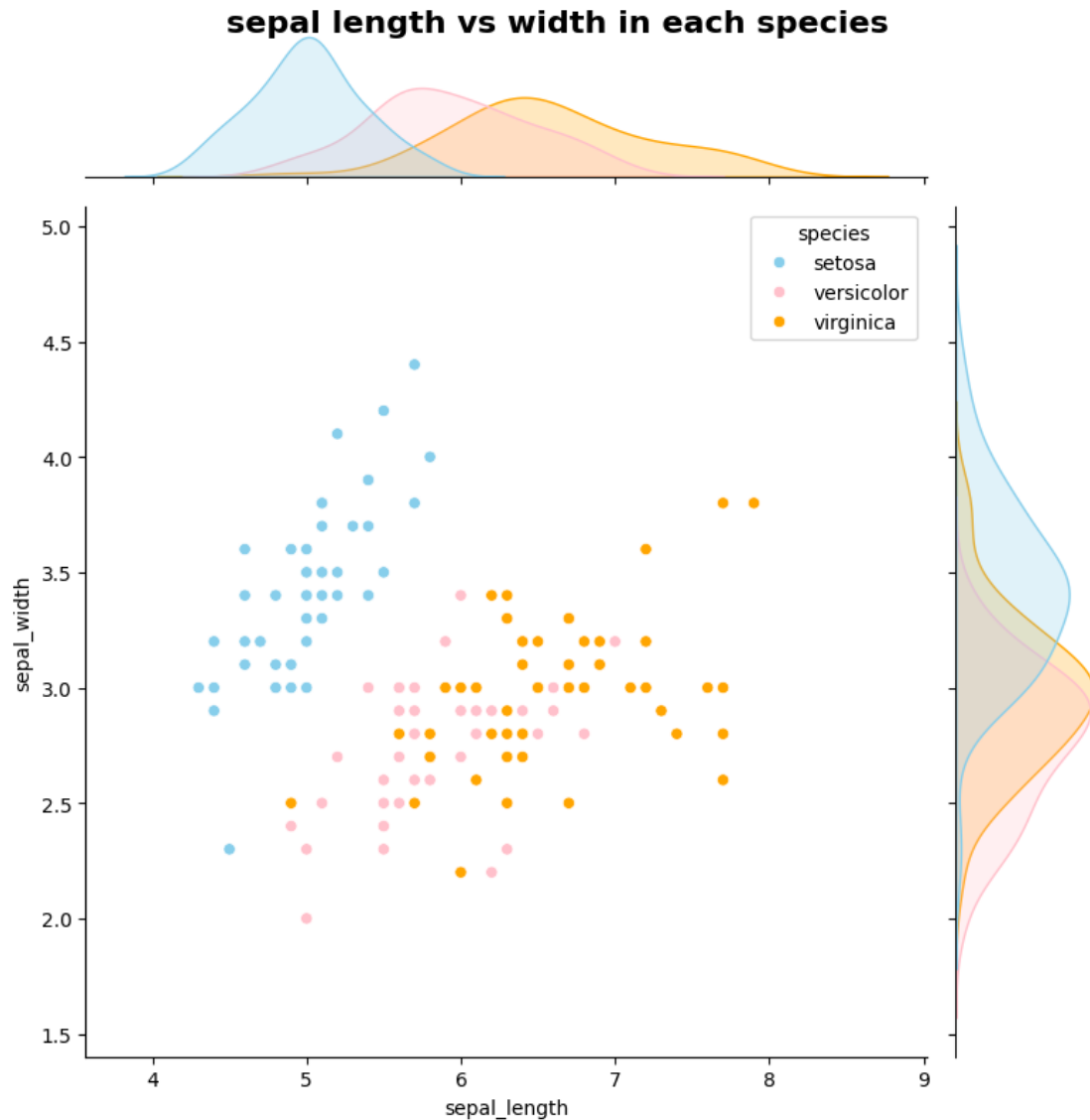
```
[26]: # Creating a figure with subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))
# Flatten the 2D array of axes for easy iteration
axes = axes.flatten()
# Feature names
features = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
# Plotting histograms for each feature using a loop
for i, feature in enumerate(features):
    sns.histplot(iris[feature], kde=True, ax=axes[i], color='skyblue',
    edgecolor='b')
    axes[i].set_title(f'Distribution of {features[i]}')
    axes[i].set_xlabel(features[i])
    axes[i].set_ylabel('Frequency')
# Adjust layout
plt.tight_layout()
plt.show()
```



## 1.6 5. Jointplot of Sepal Length vs Sepal Width:

- Write a Python program to create a joint plot to describe the individual distributions on the same plot between sepal length and sepal width.

```
[27]: # plotting a joint plot that describes
# individual distributions b/w sepal length and width in each species on the
      ↪ same plot
sns.
      ↪ jointplot(x='sepal_length',y='sepal_width',hue='species',data=iris,kind='scatter',height=8.
      ↪ 0,palette=['skyblue','pink','orange'])
plt.suptitle('sepal length vs width in each species',y=1,fontsize=16,
      ↪ fontweight='bold')
plt.show()
```



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

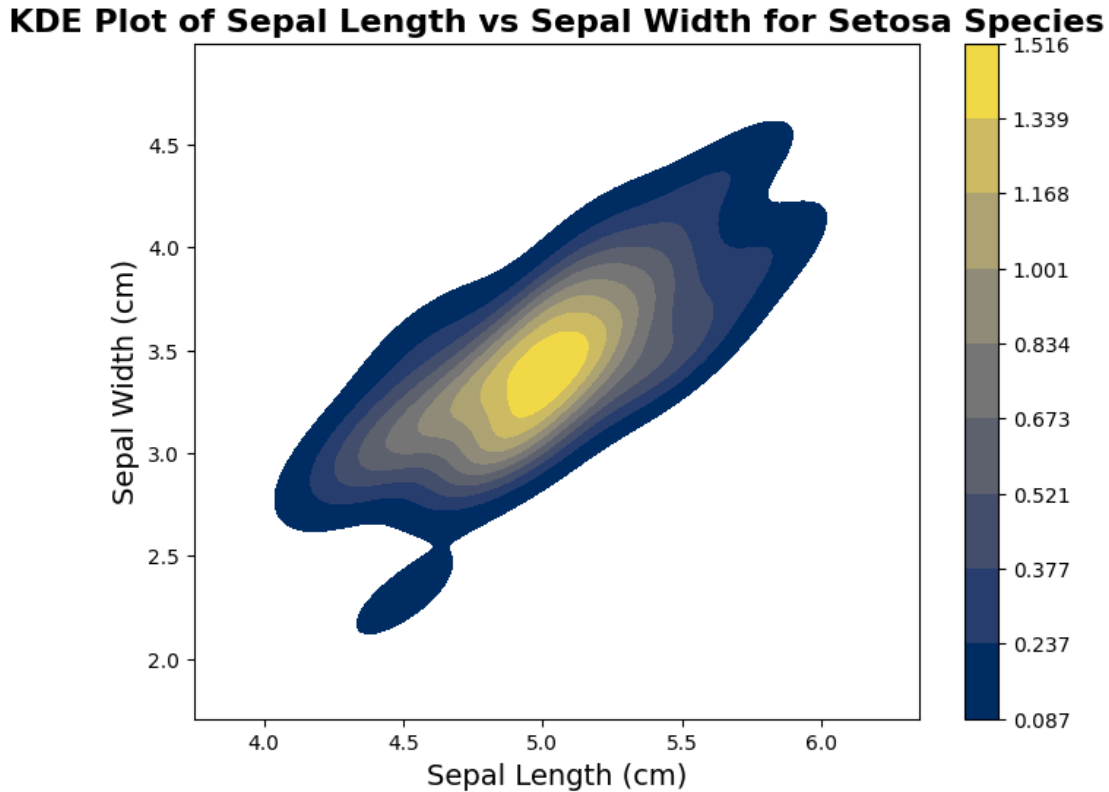
- Write a Python program using seaborn to create a KDE (Kernel Density Estimate) plot of sepal length versus sepal width for the setosa species of the Iris dataset.

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

# Create the KDE plot
plt.figure(figsize=(8, 6))
sns.kdeplot(data=setosa, x='sepal_length', y='sepal_width',
            cmap='cividis', cbar=True, fill=True)
```

```
# Add titles and labels
plt.title('KDE Plot of Sepal Length vs Sepal Width for Setosa Species',
         ↪fontsize=16, fontweight='bold')
plt.xlabel('Sepal Length (cm)', fontsize=14)
plt.ylabel('Sepal Width (cm)', fontsize=14)

# Show the plot
plt.show()
```



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

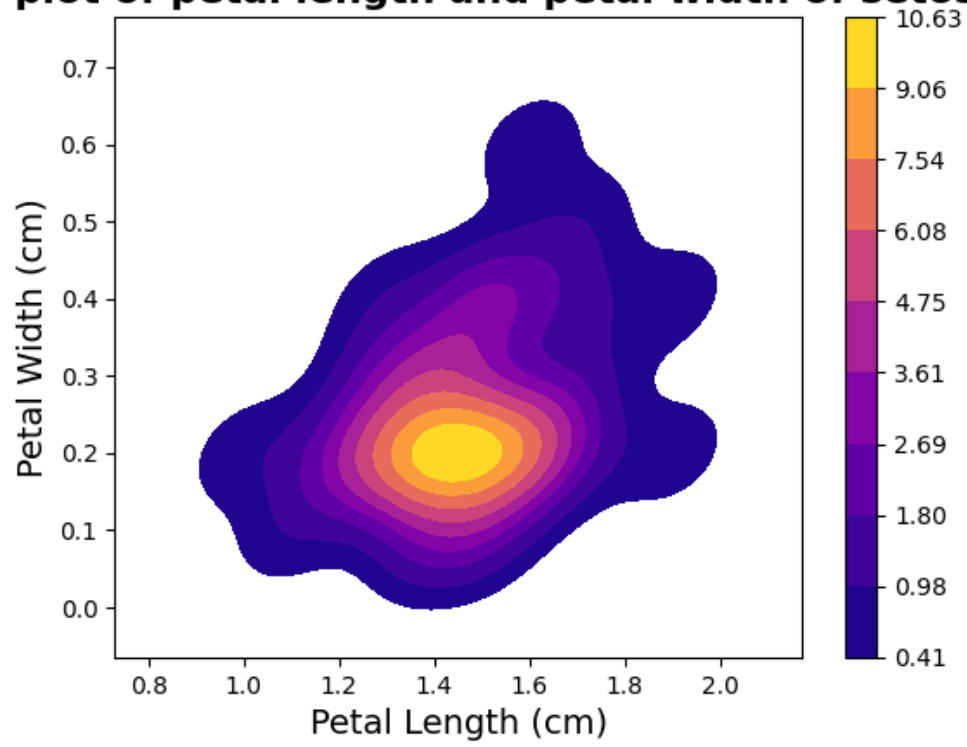
- Write a Python program using seaborn to create a KDE plot of petal length versus petal width for the setosa species.

```
[32]: setosa=iris[iris['species']=='setosa']
sns.kdeplot(x='petal_length',y='petal_width',data=setosa, cbar=True,cmap =
         ↪"plasma",fill=True)
plt.title('KDE plot of petal length and petal width of setosa', fontsize=16,
         ↪fontweight='bold')
plt.xlabel('Petal Length (cm)', fontsize=14)
plt.ylabel('Petal Width (cm)', fontsize=14)
```



```
plt.show()
```

**KDE plot of petal length and petal width of setosa**



```
[ ]:
```