Iris Flower Dataset

#

Introduction

In Machine Learning and Data Science Exploratory Data Analysis is the process of examining a data set and summarizing its main characteristics about it. It may include visual methods to better represent those characteristics or have a general understanding of the dataset. It is a very essential step in a Data Science lifecycle, often consuming a certain time.

In this article, we are going to see some of the characteristics of the Iris dataset through Exploratory Data Analysis.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [3]:
```
data=pd.read_csv('IRIS.CSV')
```

In [5]:
```
data.info
```

Out[5]:
```
<bound method DataFrame.info of      sepal_length  sepal_width  petal_length  petal_widt
h         species
0             5.1          3.5           1.4          0.2     Iris-setosa
1             4.9          3.0           1.4          0.2     Iris-setosa
2             4.7          3.2           1.3          0.2     Iris-setosa
3             4.6          3.1           1.5          0.2     Iris-setosa
4             5.0          3.6           1.4          0.2     Iris-setosa
..            ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]>
```

In [6]:
```
data.shape
```

Out[6]:
```
(150, 5)
```

In [8]:
```
data.tail()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

In [9]:
```
data.head()
```

Out[9]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [10]:
```
    data.describe()
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

In [11]:

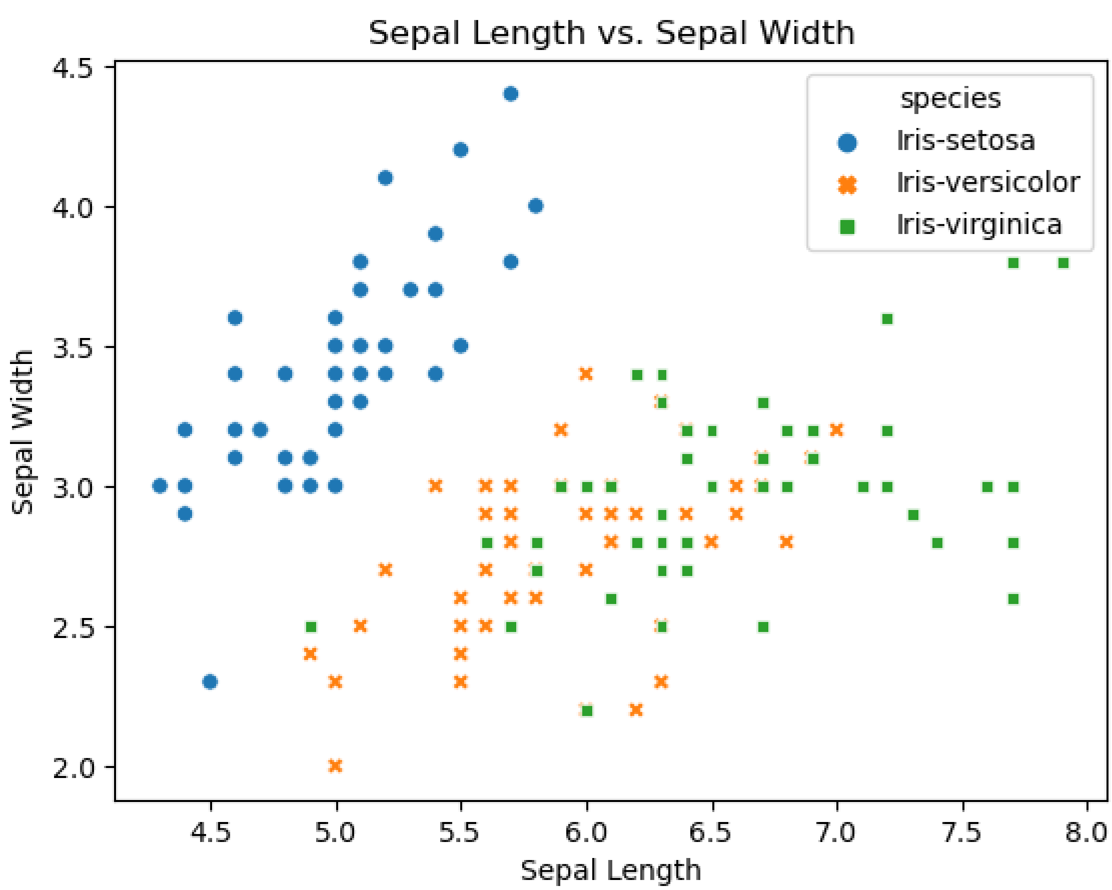
```
list(data)
```

Out[11]:

```
['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species']
```

In [23]:

```
sns.scatterplot(data=data, x='sepal_length', y='sepal_width',color = 'g', hue='species',
plt.title('Sepal Length vs. Sepal Width')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



In [24]:

```
list(data)
```

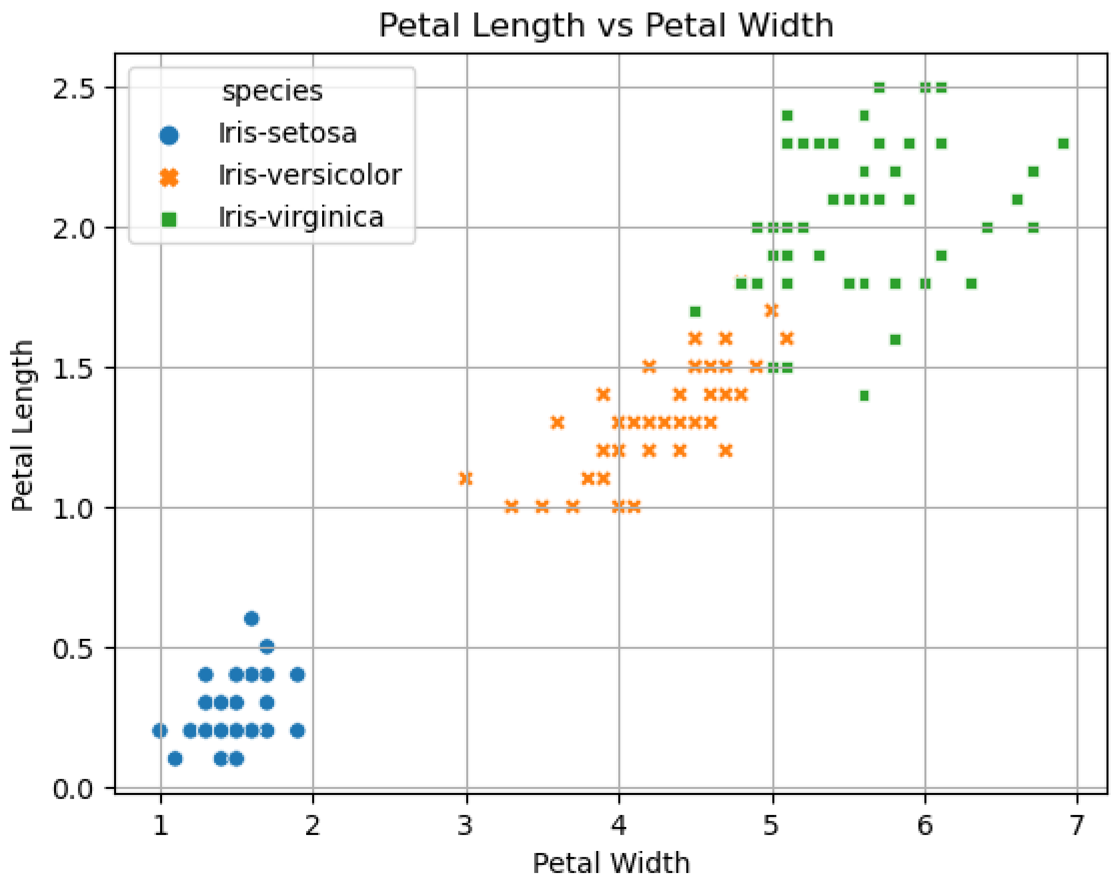
Out[24]:

```
['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species']
```

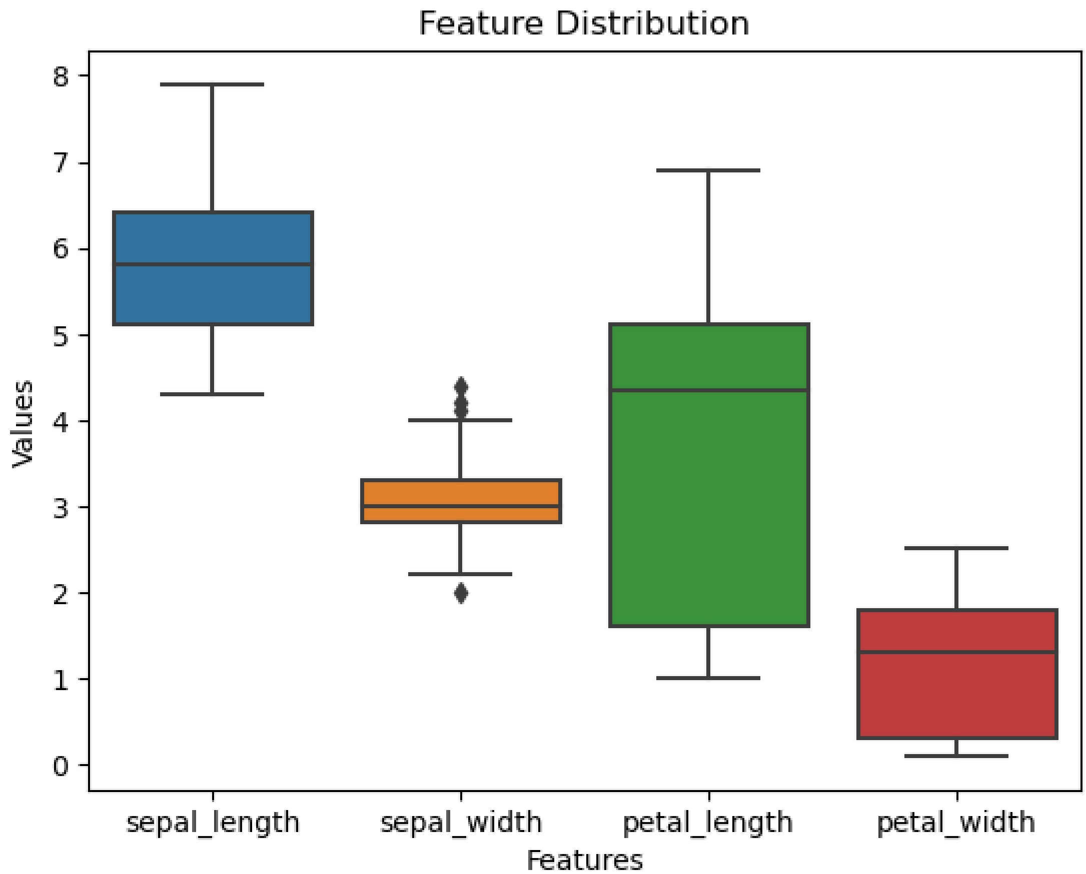
In [37]:

```
sns.scatterplot(data=data, x='petal_length', y ='petal_width', hue = 'species', style ='
plt.title('Petal Length vs Petal Width ')
plt.xlabel('Petal Width')
plt.ylabel('Petal Length')
```

```
plt.grid(True)
plt.show()
```



In [39]:
```
sns.boxplot(data=data.drop('species', axis=1))
plt.title('Feature Distribution')
plt.xlabel('Features')
plt.ylabel('Values')
plt.show()
```



In [43]:
```
features = data.drop('species', axis=1)
target = data['species']
label_encoder = LabelEncoder()
target_encoded = label_encoder.fit_transform(target)
```

In [44]:
```
X_train, X_test, y_train, y_test = train_test_split(features, target_encoded, test_size=
```

In [45]:
```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[45]:
```
▼        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [46]:
```
y_pred = model.predict(X_test)
y_pred
```

Out[46]:
```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```

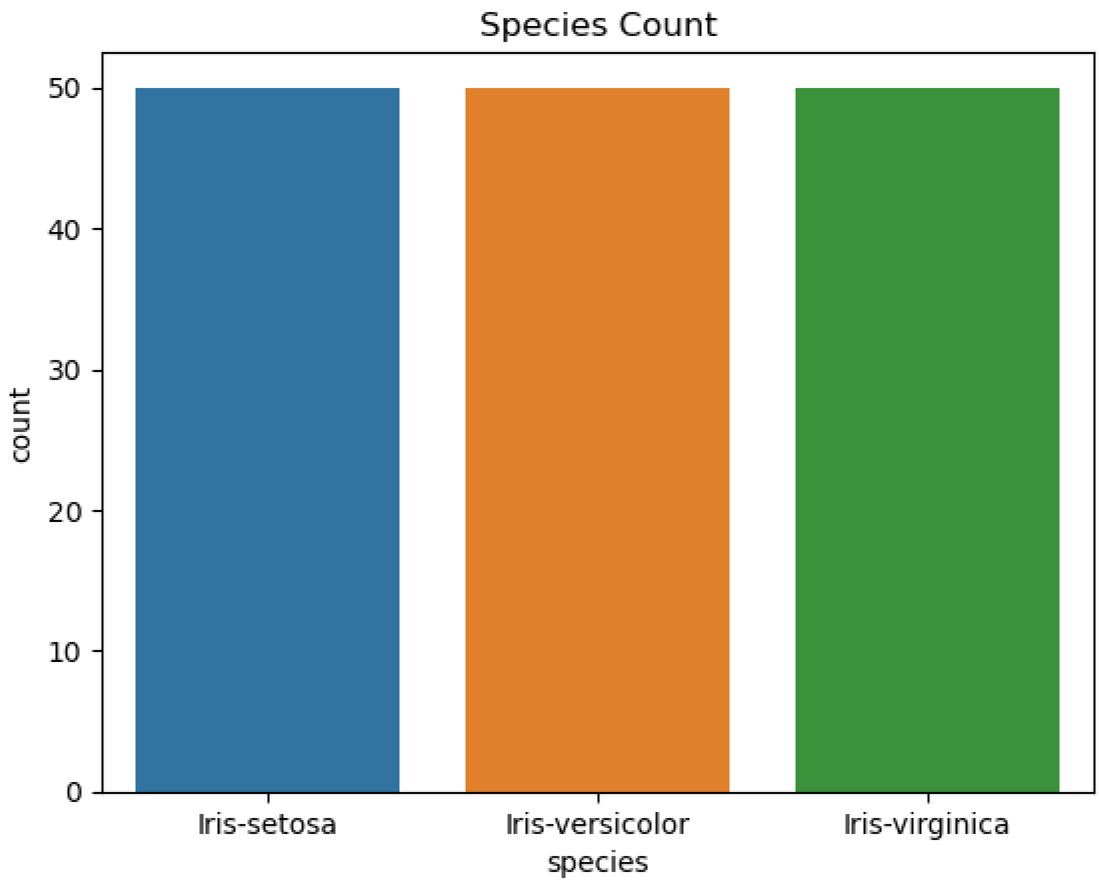
In [48]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

In [49]:
```
input_data = np.array([[5.1, 3.5, 1.4, 0.2]])  # Example data for prediction
predicted_class = model.predict(input_data)[0]
predicted_species = label_encoder.inverse_transform([predicted_class])[0]
print("Predicted Species:", predicted_species)
```

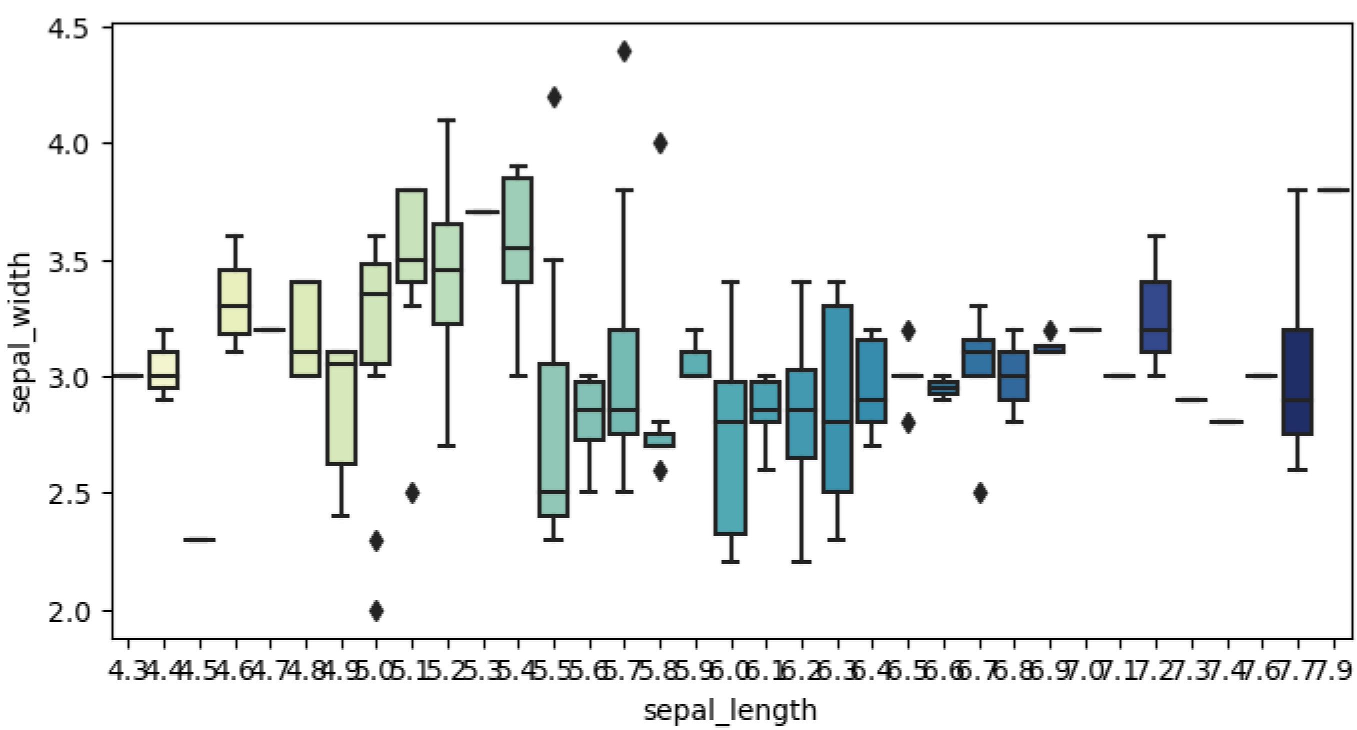
```
Predicted Species: Iris-setosa
C:\ProgramData\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not
have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

In [59]:
```
sns.countplot (x='species', data=data)
plt.title('Species Count')

plt.show()
```



In [71]:
```
## Box plot
plt.figure(figsize=(8,4))
sns.boxplot(x='sepal_length',y='sepal_width',data=data ,palette='YlGnBu')
```

Out[71]: <Axes: xlabel='sepal_length', ylabel='sepal_width'>

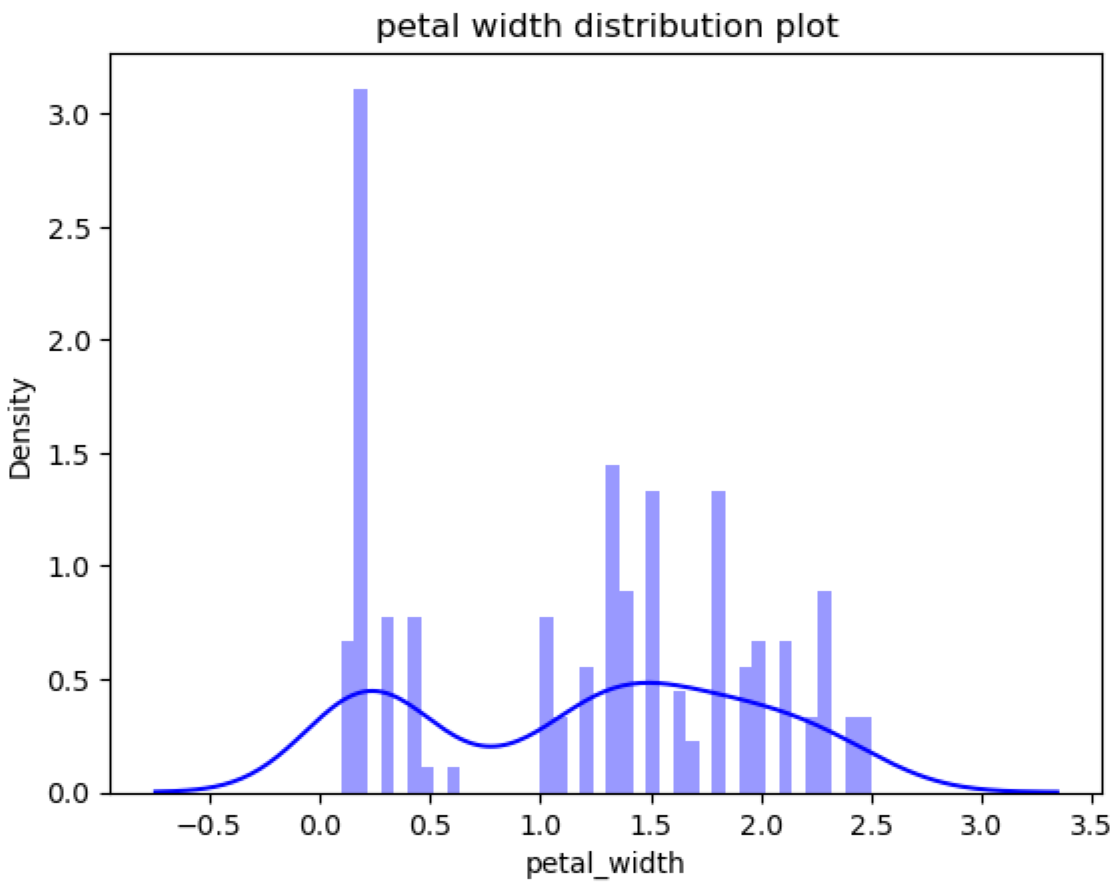


Loading [MathJax]/extensions/Safe.js

In [69]:
```
## Distribution of particular species
sns.distplot(a=data['petal_width'], bins=40, color='b')
plt.title('petal width distribution plot')
```

```
C:\Users\Ajeeth S\AppData\Local\Temp\ipykernel_27244\3355427628.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(a=data['petal_width'], bins=40, color='b')
```

Out[69]:
```
Text(0.5, 1.0, 'petal width distribution plot')
```



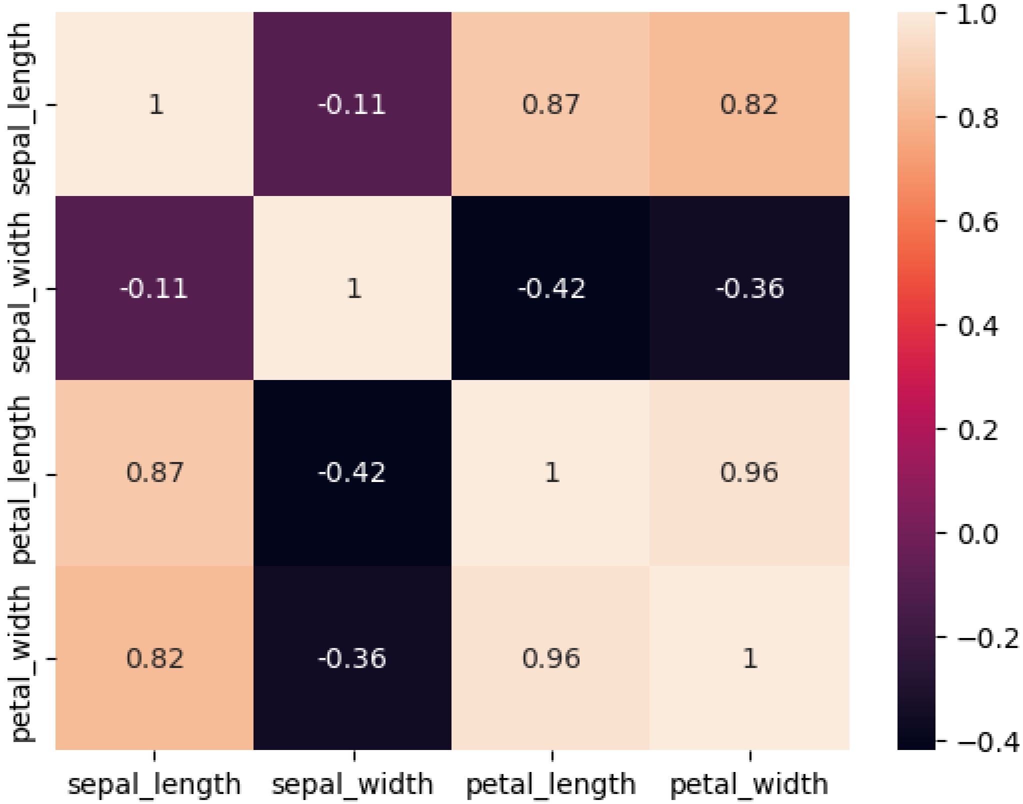
In [ ]:

In [74]:
```
sns.heatmap(data.corr(), linecolor = 'red' , annot= True)
```

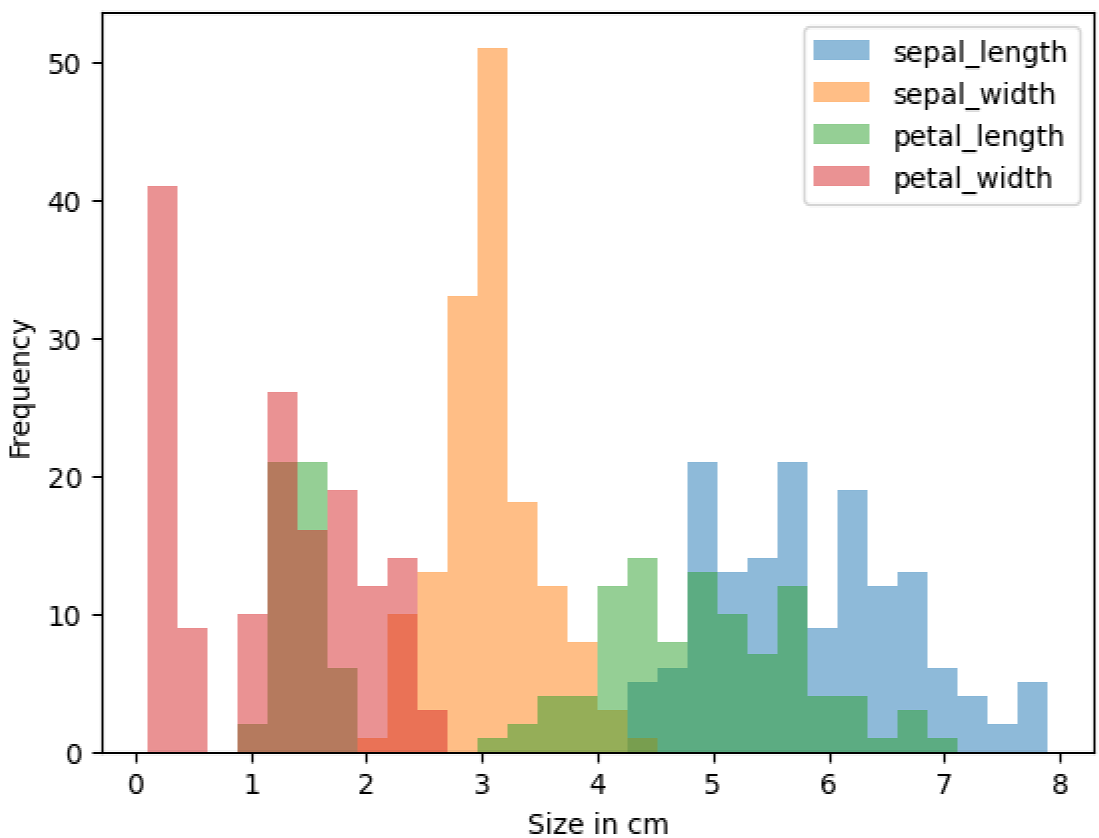
```
C:\Users\Ajeeth S\AppData\Local\Temp\ipykernel_27244\2957642093.py:1: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
ill default to False. Select only valid columns or specify the value of numeric_only to
silence this warning.
  sns.heatmap(data.corr(), linecolor = 'red' , annot= True)
```

Out[74]:
```
<Axes: >
```

Loading [MathJax]/extensions/Safe.js



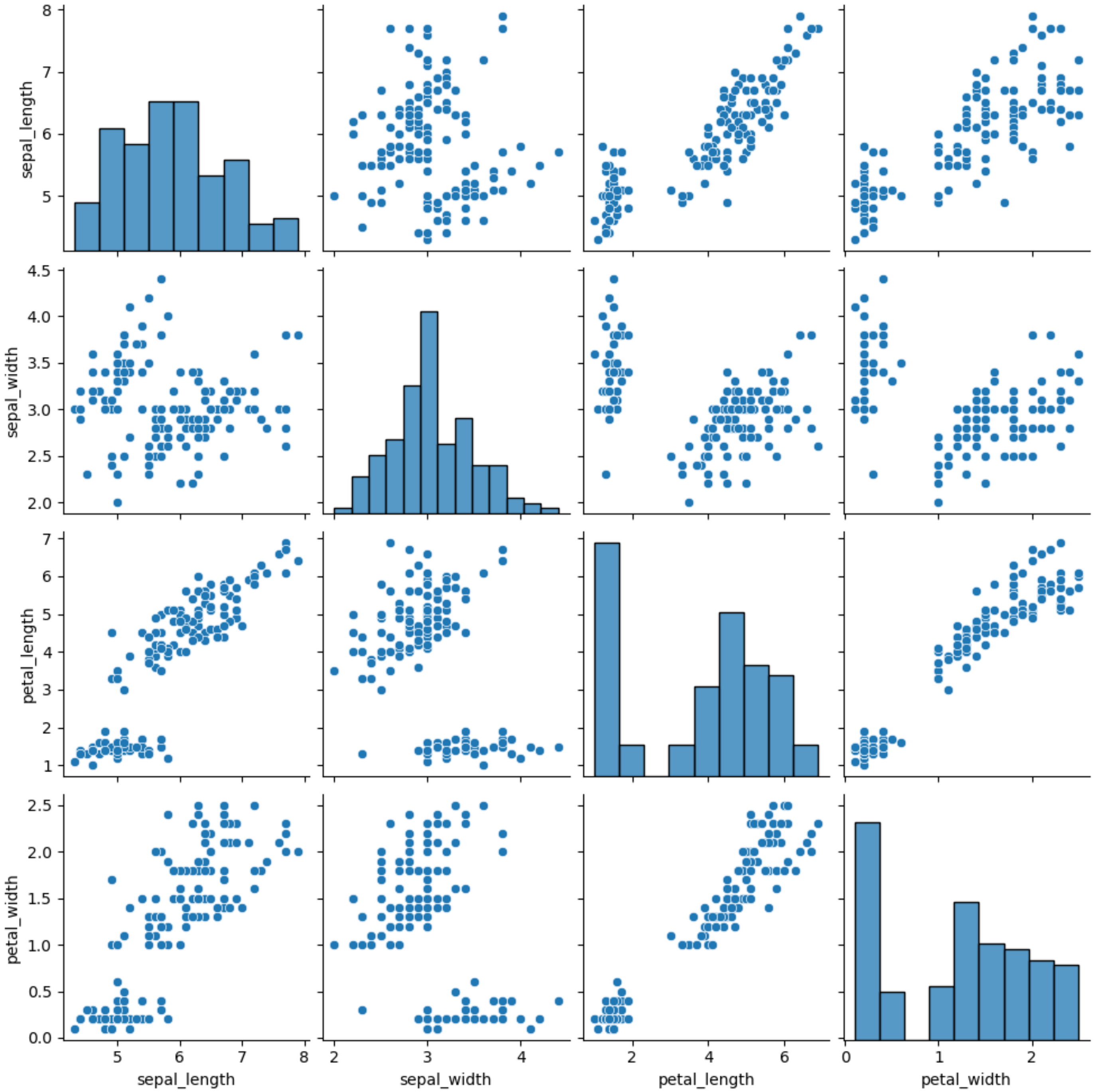
In [75]:
```
axis = data.plot.hist(bins=30, alpha=0.5)
axis.set_xlabel('Size in cm');
```



In [80]:
```
sns.pairplot(data)
```

Out[80]: <seaborn.axisgrid.PairGrid at 0x11559c6e230>

Loading [MathJax]/extensions/Safe.js



Conclusion

Exploratory Data Analysis is extremely used by both Data Scientists and Analysts. It tells a lot about the characteristics of the given data, its distribution, and how it can be useful.

In [ ]: