In [1]:

```
import pandas as pd

# Reading the CSV file
df = pd.read_csv("iris_csv.csv")

# Printing top 5 rows
df.head()
```

Out[1]:

	sepallength	sepalwidth	petallength	petalwidth	class
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 [2]:

```
df.shape
```

Out[2]:

```
(150, 5)
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   sepallength  150 non-null    float64
 1   sepalwidth   150 non-null    float64
 2   petallength  150 non-null    float64
 3   petalwidth   150 non-null    float64
 4   class        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [4]:

```
df.describe()
```

Out[4]:

	sepallength	sepalwidth	petallength	petalwidth
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 [38]:

```
df.tail()
```

Out[38]:

	sepallength	sepalwidth	petallength	petalwidth	class
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 [5]:

```
# Checking Missing Values
df.isnull().sum()
```

Out[5]:

```
sepallength    0
sepalwidth     0
petallength    0
petalwidth     0
class          0
dtype: int64
```

In [6]:

```
# Checking Duplicates
data = df.drop_duplicates(subset ="class",)
data
```

Out[6]:

	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
50	7.0	3.2	4.7	1.4	Iris-versicolor
100	6.3	3.3	6.0	2.5	Iris-virginica

In [7]:

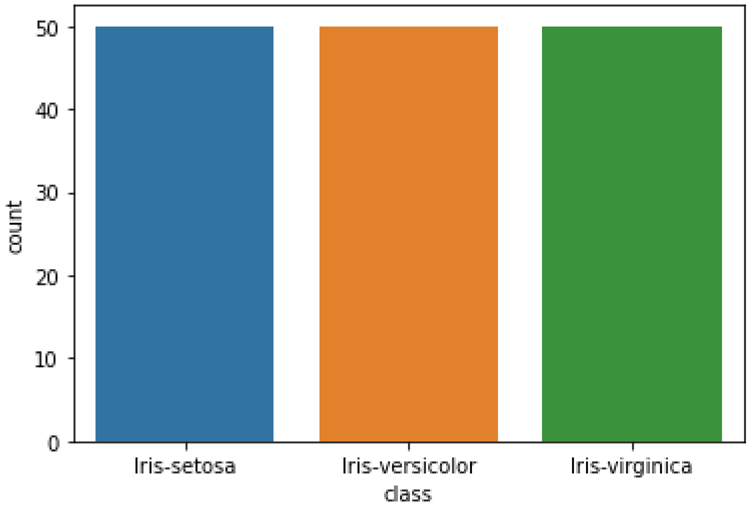
```
df.value_counts("class")
```

Out[7]:

```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

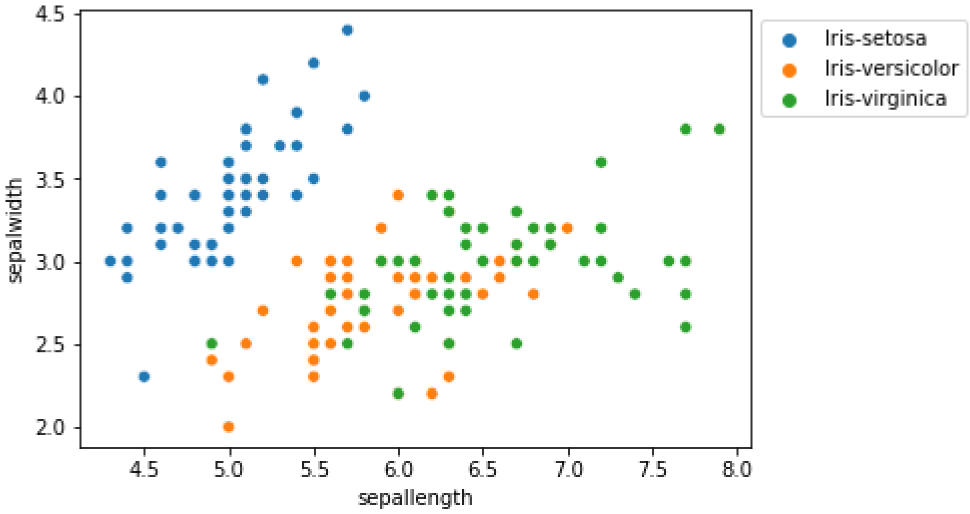
In [8]:

```
# Data Visualization
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt

# countplot for class
sns.countplot(x='class', data=df, )
plt.show()
```



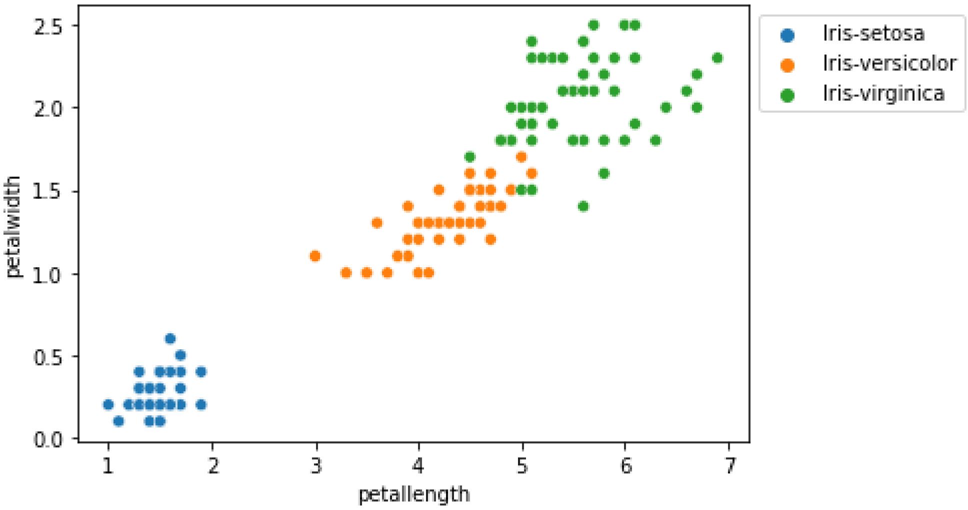
In [9]:

```
# Relation between variables
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.scatterplot(x='sepallength', y='sepalwidth',
                hue='class', data=df, )

# Placing Legend outside the Figure
plt.legend(bbox_to_anchor=(1, 1), loc=2)

plt.show()
```



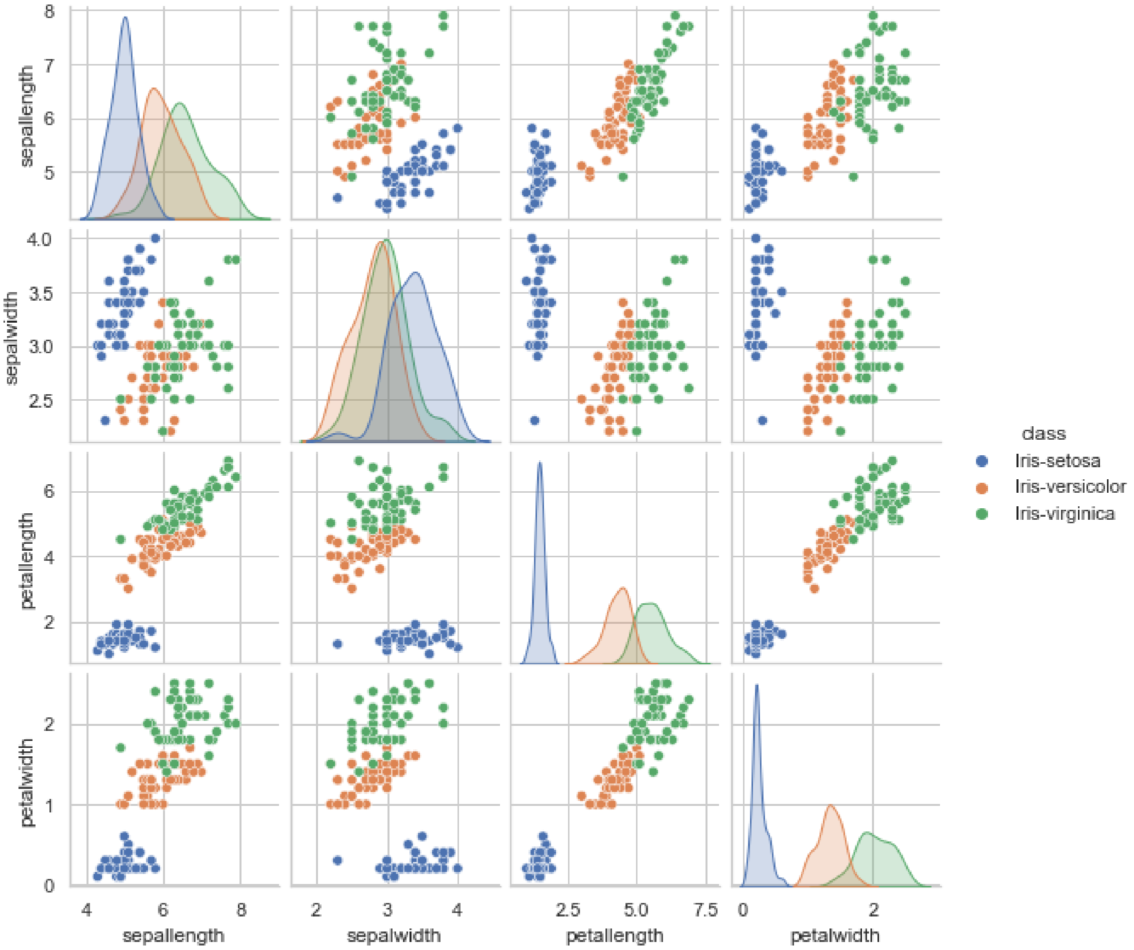
In [10]:

```
# Comparing Petal Length and Petal Width
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.scatterplot(x='petallength', y='petalwidth',
                hue='class', data=df, )

# Placing Legend outside the Figure
plt.legend(bbox_to_anchor=(1, 1), loc=2)

plt.show()
```



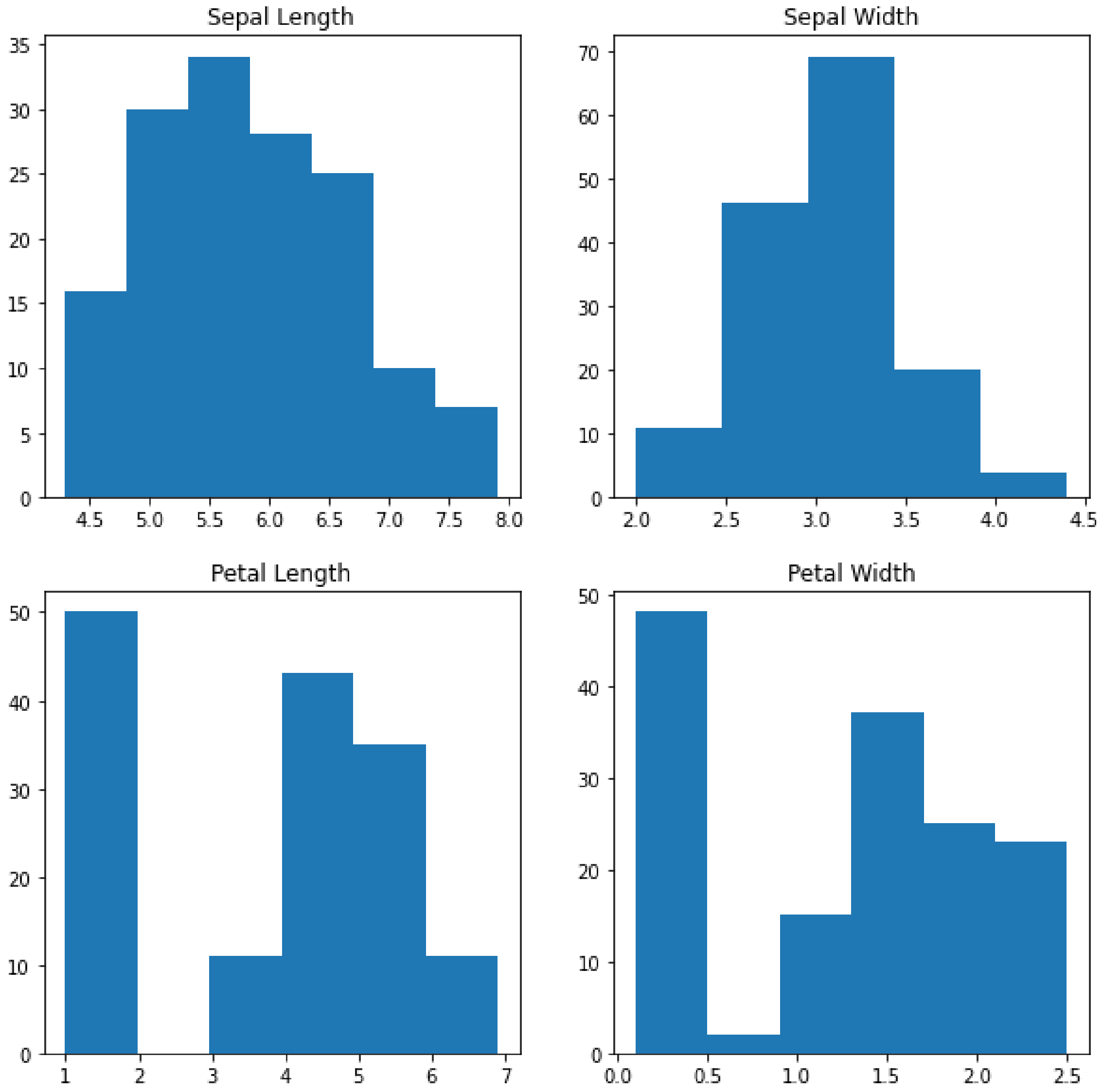
In [51]:
```
# multivariate analysis using a pairplot
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.pairplot(df, hue='class', height=2)
```

Out[51]:
```
<seaborn.axisgrid.PairGrid at 0x242cab26f40>
```



In [12]:

```
# Histogram is used for uni as well as bi-variate analysis
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


fig, axes = plt.subplots(2, 2, figsize=(10,10))

axes[0,0].set_title("Sepal Length")
axes[0,0].hist(df['sepallength'], bins=7)

axes[0,1].set_title("Sepal Width")
axes[0,1].hist(df['sepalwidth'], bins=5);

axes[1,0].set_title("Petal Length")
axes[1,0].hist(df['petallength'], bins=6);

axes[1,1].set_title("Petal Width")
axes[1,1].hist(df['petalwidth'], bins=6);
```


Sepal Length

Sepal Width

Petal Length

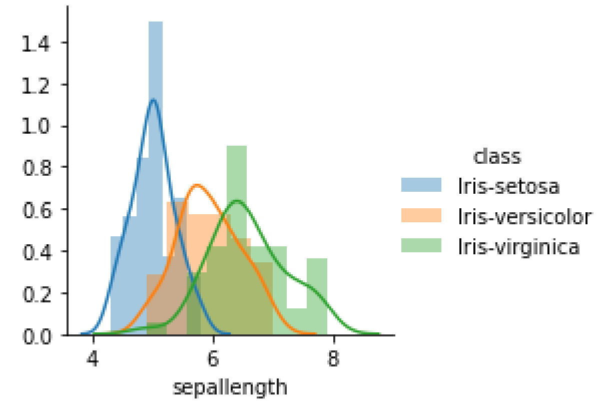
Petal Width

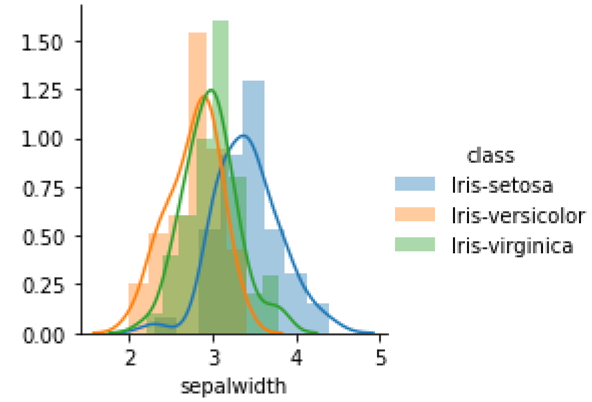
In [13]:

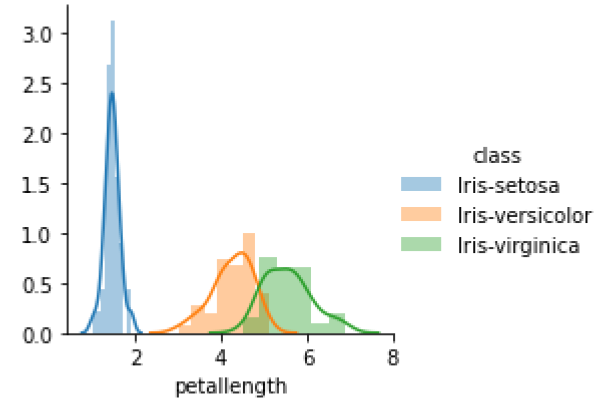
```
# Histograms with Distplot Plot
# importing packages


plot = sns.FacetGrid(df, hue="class")
plot.map(sns.distplot, "sepallength").add_legend()

plot = sns.FacetGrid(df, hue="class")
plot.map(sns.distplot, "sepalwidth").add_legend()

plot = sns.FacetGrid(df, hue="class")
plot.map(sns.distplot, "petallength").add_legend()

plot = sns.FacetGrid(df, hue="class")
plot.map(sns.distplot, "petalwidth").add_legend()

plt.show()
```

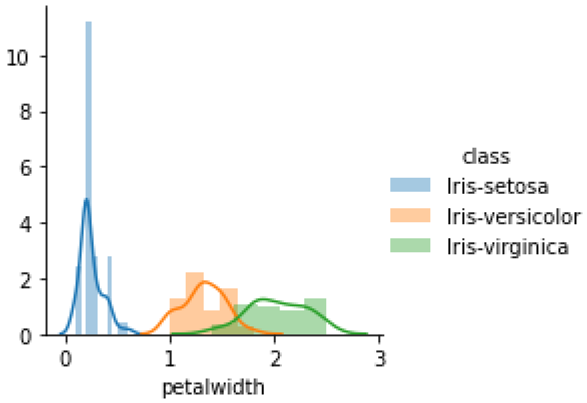
```
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
```

```
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Chiranjeevi\anaconda3\lib\site-packages\seaborn\distributions.py:
2557: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-lev
el function for histograms).
  warnings.warn(msg, FutureWarning)
```









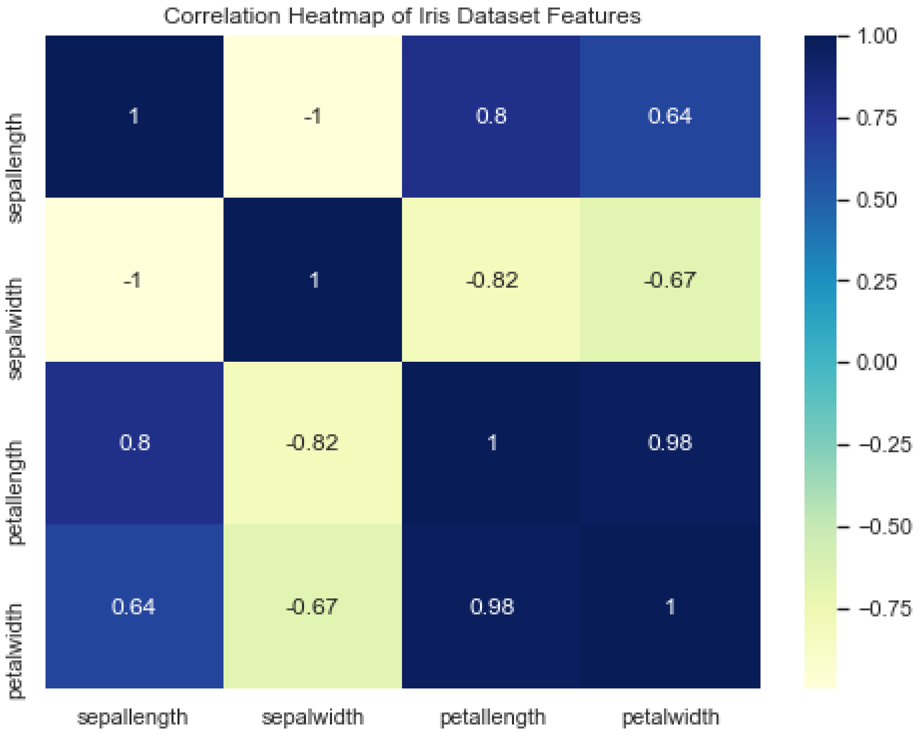
```
In [14]: # Handling Correlation
         data.corr(method='pearson')
```

Out[14]:

	sepallength	sepalwidth	petallength	petalwidth
sepallength	1.000000	-0.999226	0.795795	0.643817
sepalwidth	-0.999226	1.000000	-0.818999	-0.673417
petallength	0.795795	-0.818999	1.000000	0.975713
petalwidth	0.643817	-0.673417	0.975713	1.000000

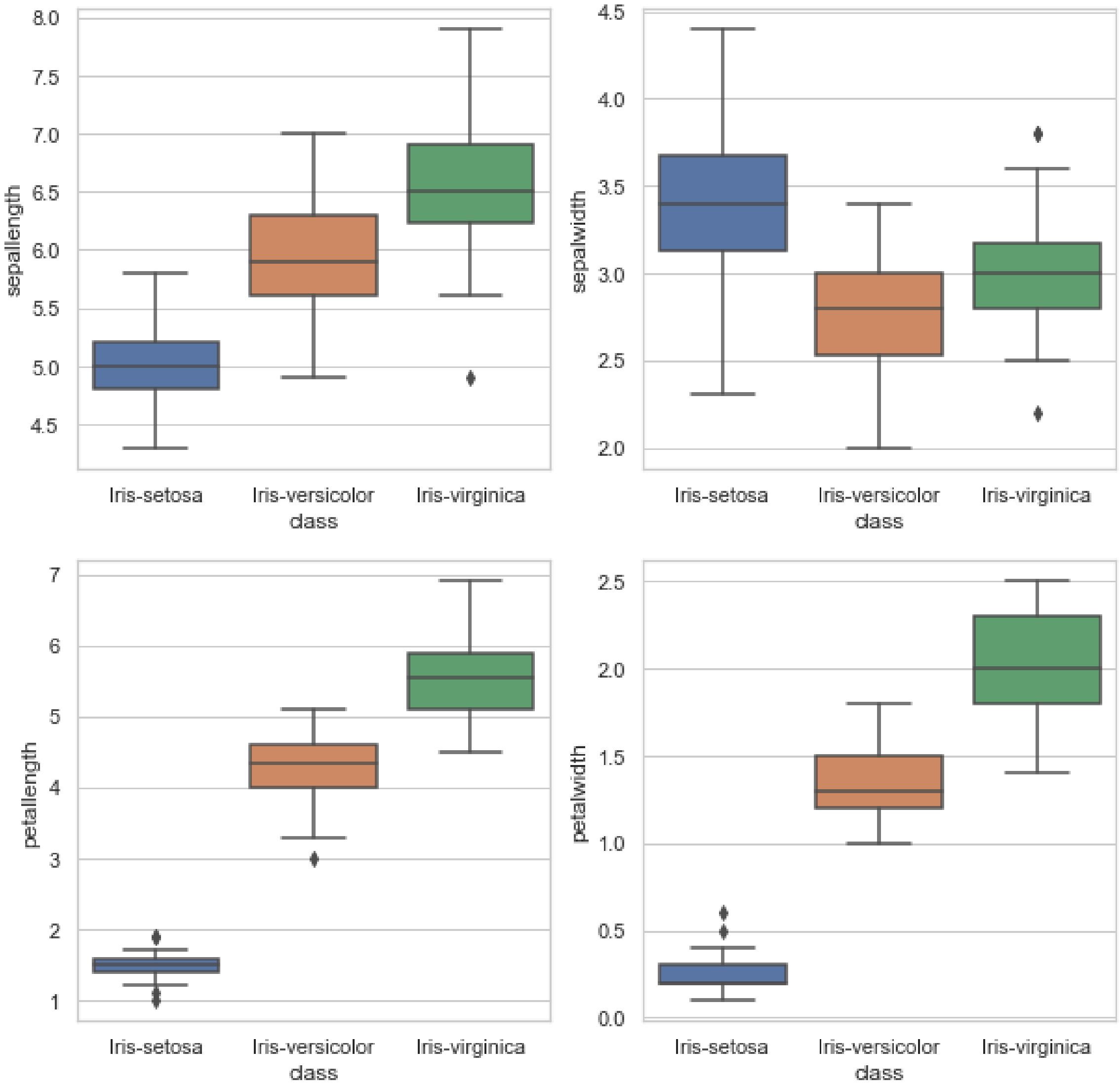
In [26]:

```
# Heatmaps
# plot the above-found correlation using the heatmaps.
# importing packages
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix
feature_names = data.corr(method='pearson')

# Create a heatmap using Seaborn
sns.set(style="whitegrid")
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="YlGnBu", xticklabels=featu
plt.title('Correlation Heatmap of Iris Dataset Features')
plt.show()
```

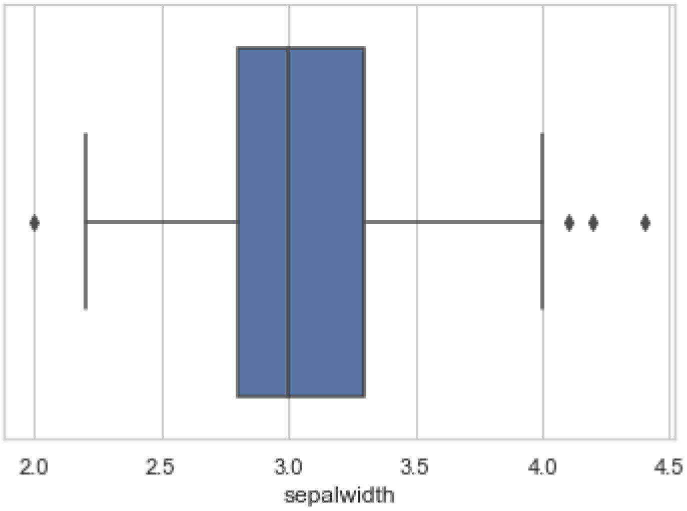


In [27]:

```
# Box plots
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt

def graph(y):
    sns.boxplot(x="class", y=y, data=df)

plt.figure(figsize=(10,10))

# Adding the subplot at the specified
# grid position
plt.subplot(221)
graph('sepallength')

plt.subplot(222)
graph('sepalwidth')

plt.subplot(223)
graph('petallength')

plt.subplot(224)
graph('petalwidth')

plt.show()
```



```
In [30]: # Handling Outliers

         sns.boxplot(x='sepalwidth', data=df)

Out[30]: <AxesSubplot:xlabel='sepalwidth'>
```



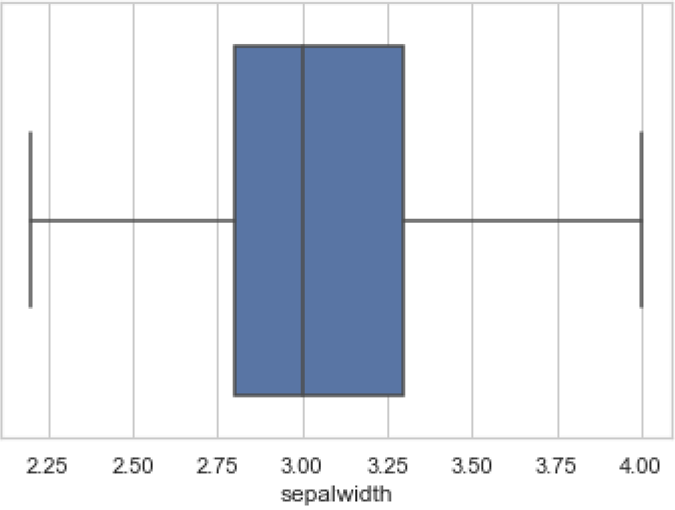
In [34]:

```
# Removing Outliers
import numpy as np
Q1 = np.percentile(df['sepalwidth'], 25,
                interpolation = 'midpoint')

Q3 = np.percentile(df['sepalwidth'], 75,
                interpolation = 'midpoint')
IQR = Q3 - Q1

print("Old Shape: ", df.shape)

# Upper bound
upper = np.where(df['sepalwidth'] >= (Q3+1.5*IQR))

# Lower bound
lower = np.where(df['sepalwidth'] <= (Q1-1.5*IQR))

# Removing the Outliers
df.drop(upper[0], inplace = True)
df.drop(lower[0], inplace = True)

print("New Shape: ", df.shape)

sns.boxplot(x='sepalwidth', data=df)
```

```
Old Shape:  (146, 5)
New Shape:  (146, 5)

<ipython-input-34-dfcb10346225>:3: DeprecationWarning: the `interpolation=
` argument to percentile was renamed to `method=`, which has additional op
tions.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encoura
ged to review the method they. (Deprecated NumPy 1.22)
  Q1 = np.percentile(df['sepalwidth'], 25,
<ipython-input-34-dfcb10346225>:6: DeprecationWarning: the `interpolation=
` argument to percentile was renamed to `method=`, which has additional op
tions.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encoura
ged to review the method they. (Deprecated NumPy 1.22)
  Q3 = np.percentile(df['sepalwidth'], 75,
```

Out[34]:

```
<AxesSubplot:xlabel='sepalwidth'>
```



In [42]:

```
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

y
```

Out[42]:

```
0          Iris-setosa
1          Iris-setosa
2          Iris-setosa
3          Iris-setosa
4          Iris-setosa
              ...
145     Iris-virginica
146     Iris-virginica
147     Iris-virginica
148     Iris-virginica
149     Iris-virginica
Name: class, Length: 146, dtype: object
```

In [46]:

```
from sklearn.model_selection import train_test_split

x1,x2,y1,y2 = train_test_split(X,y, test_size=0.2, random_state=42)
```

In [47]:

```
x1.shape, x2.shape, y1.shape, y2.shape
```

Out[47]:

```
((116, 4), (30, 4), (116,), (30,))
```

In [48]:

```
from sklearn.svm import SVC

svc = SVC()
svc.fit(x1,y1)
```

Out[48]:

```
▾ SVC

SVC()
```

In [50]:

```
svc.score(x2,y2)*100
```

Out[50]:

```
96.66666666666667
```