In [3]:

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

In [10]:

```
df = pd.read_csv('C:/Users/User/Downloads/archive/IRIS.csv')

df_c = df.copy()
df
```

Out[10]:

	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
...	...	...	...	...	...
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 × 5 columns

sepal - sepals are modified leaves that encase tha developing flower. Usuallu green in color.

petals - the colores part of flower.

Tasks:

1. Based on the given feature we have to classify the given species

2. comparision of their sizes

In [11]:

```
df.sample(9)
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width	species
47	4.6	3.2	1.4	0.2	Iris-setosa
90	5.5	2.6	4.4	1.2	Iris-versicolor
67	5.8	2.7	4.1	1.0	Iris-versicolor
84	5.4	3.0	4.5	1.5	Iris-versicolor
9	4.9	3.1	1.5	0.1	Iris-setosa
50	7.0	3.2	4.7	1.4	Iris-versicolor
45	4.8	3.0	1.4	0.3	Iris-setosa
146	6.3	2.5	5.0	1.9	Iris-virginica
82	5.8	2.7	3.9	1.2	Iris-versicolor

In [12]:

```
df.info()
#no null values in the data set
```

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

In [14]:

```
df.species.unique()
#different types of species avaliable
```

Out[14]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [15]:

```
df.corr()
```

Out[15]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

In [16]:

```
df.species.value_counts()
```

Out[16]:

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: species, dtype: int64
```

In [22]:

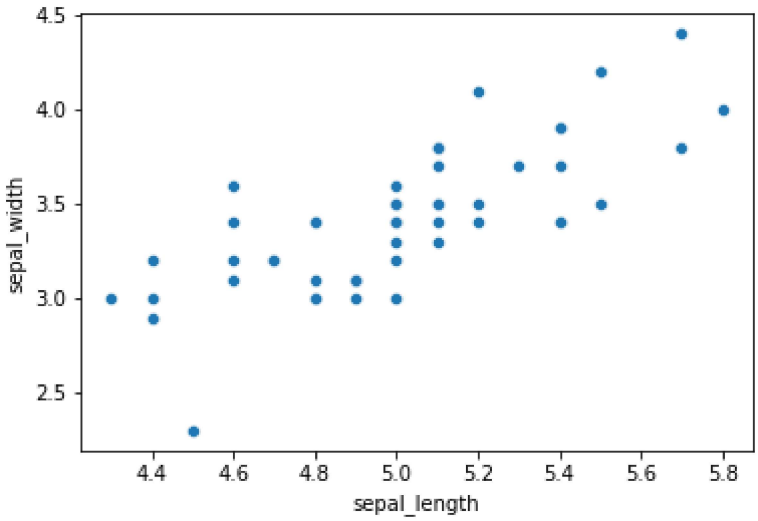
```
setos = df[df["species"] == 'Iris-setosa']
versi = df[df["species"] == 'Iris-versicolor']
virg = df[df["species"] == 'Iris-virginica']
```

In [23]:

```
sns.scatterplot(x = setos['sepal_length'],y = setos['sepal_width'])
```

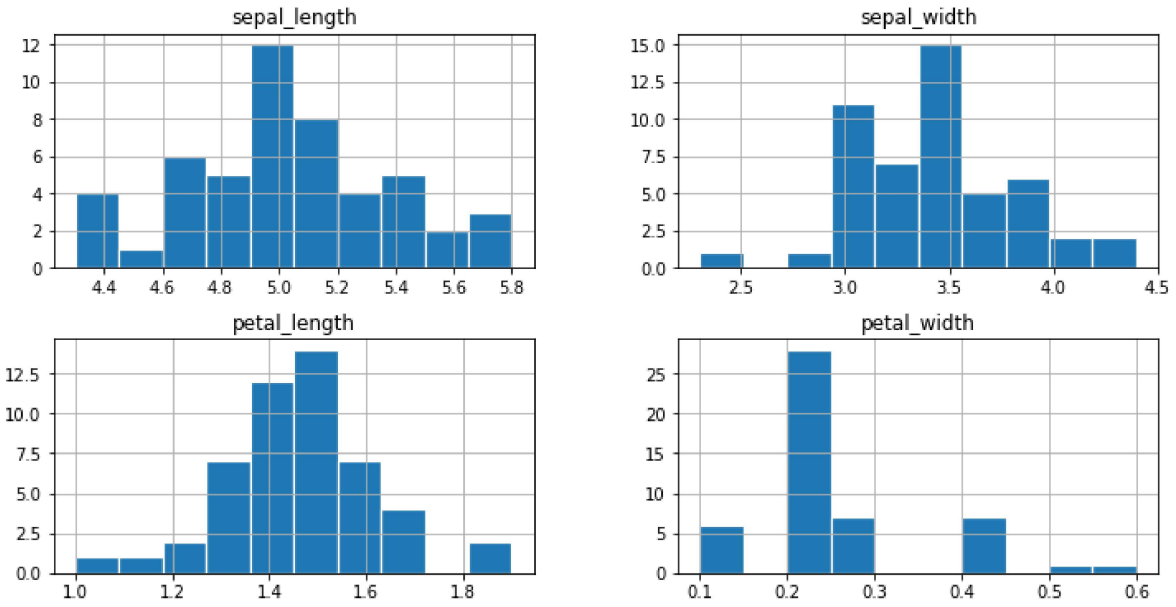
Out[23]:

```
<AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>
```



In [24]:

```
setos.hist(edgecolor='white', linewidth =1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



In [26]:

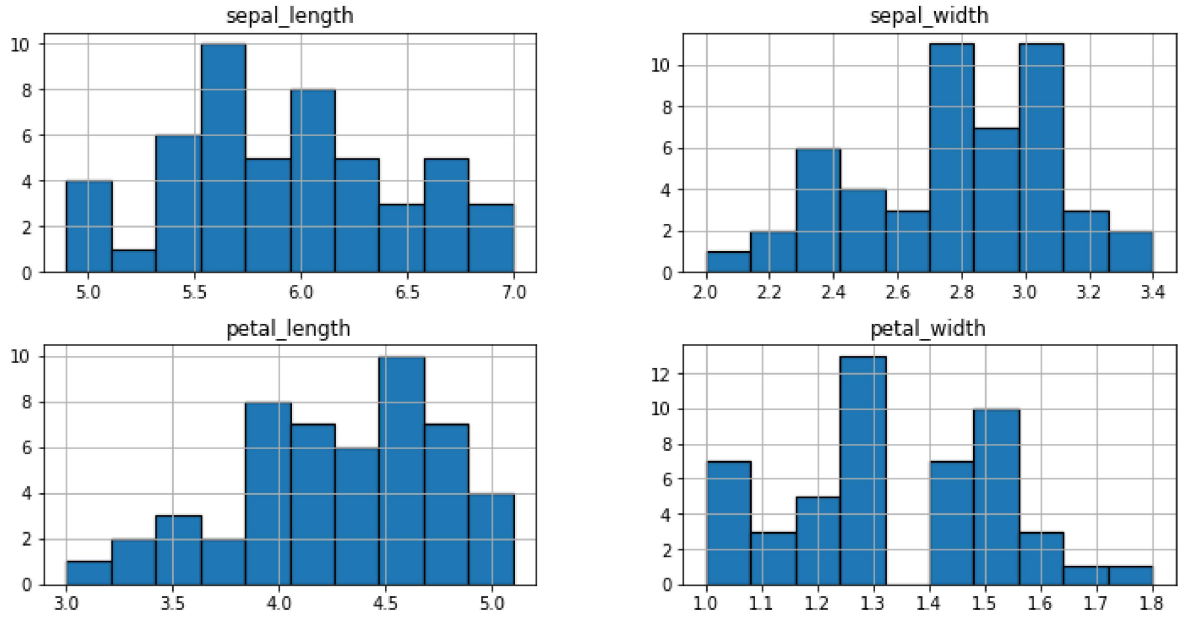
```
setos.describe()
```

Out[26]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

In [27]:

```
versi.hist(edgecolor='black', linewidth =1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



In [28]:

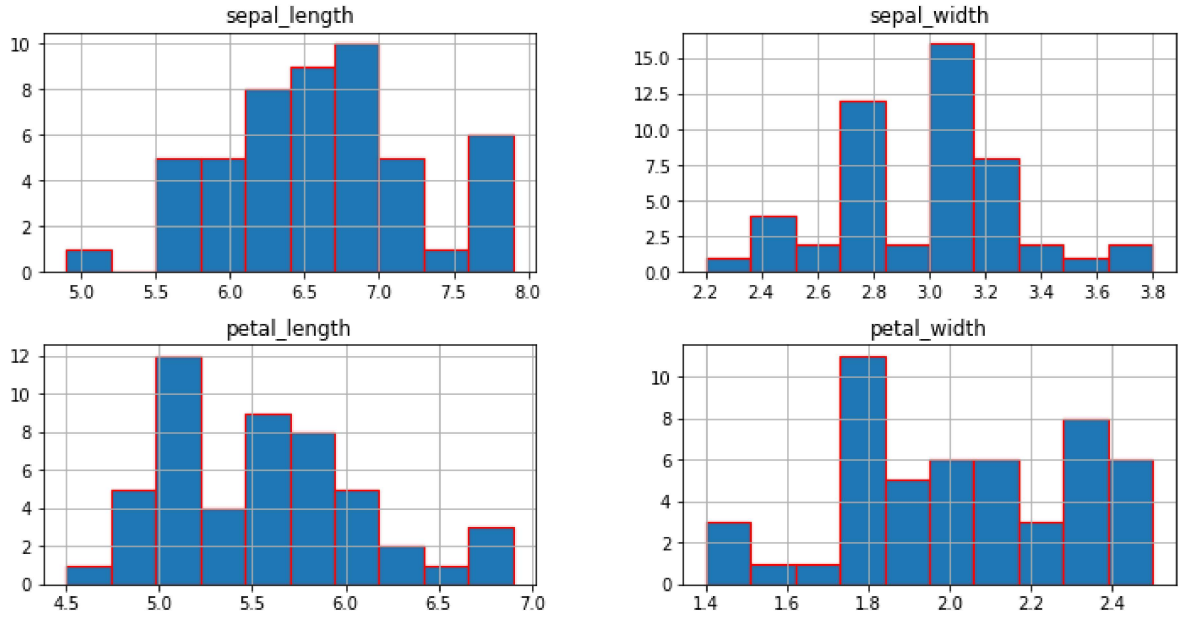
```
versi.describe()
```

Out[28]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

In [29]:

```
virg.hist(edgecolor='red', linewidth =1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



In [30]:

```
virg.describe()
```

Out[30]:

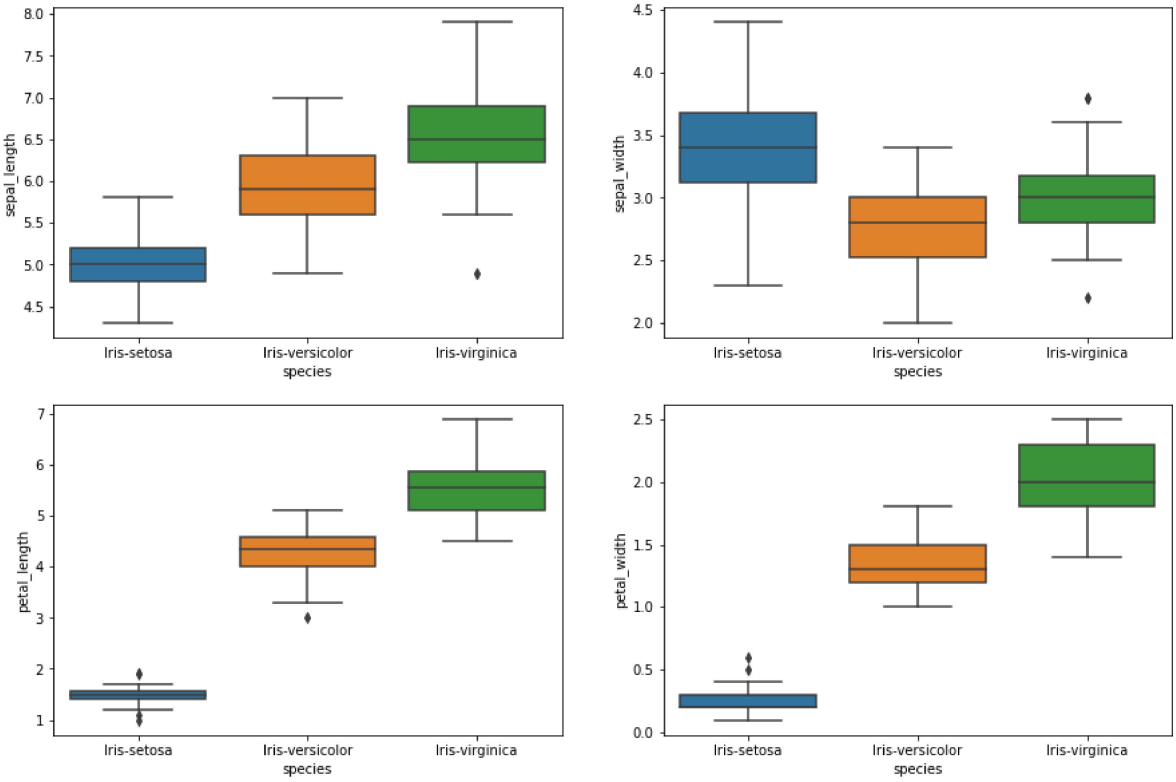
	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

In [32]:

```
plt.figure(figsize=(15,10))
plt.subplot(221)
sns.boxplot(x=df['species'],y=df['sepal_length'])
plt.subplot(222)
sns.boxplot(x=df['species'],y=df['sepal_width'])
plt.subplot(223)
sns.boxplot(x=df['species'],y=df['petal_length'])
plt.subplot(224)
sns.boxplot(x=df['species'],y=df['petal_width'])
```

Out[32]:

```
<AxesSubplot:xlabel='species', ylabel='petal_width'>
```



In [34]:

```
setos['petal_length'].mean()
setos['petal_length'].value_counts()
(setos['petal_length']<=1.2).value_counts()
setos.drop(setos.loc[setos['petal_length']<=1.2].index,inplace=True)
setos.shape
```

```
C:\Users\User\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  return super().drop(
```

Out[34]:

```
(46, 5)
```

In [35]:

```
setos[setos['petal_width']>=0.5]
```

Out[35]:

	sepal_length	sepal_width	petal_length	petal_width	species
23	5.1	3.3	1.7	0.5	Iris-setosa
43	5.0	3.5	1.6	0.6	Iris-setosa

In [37]:

```
(setos['sepal_width']>=4.2).value_counts()
```

Out[37]:

```
False    44
True      2
Name: sepal_width, dtype: int64
```

In [40]:

```
((setos['sepal_width'])>=2.8).value_counts()
```

Out[40]:

```
True     45
False     1
Name: sepal_width, dtype: int64
```

In [41]:

```
setos.drop(setos.loc[setos['petal_width']>=0.5].index,inplace=True)
setos.drop(setos.loc[setos['sepal_width']<=2.8].index,inplace=True)
setos.shape
```

```
C:\Users\User\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  return super().drop(
```

Out[41]:

```
(43, 5)
```

In [43]:

```
setos.describe()
```

Out[43]:

	sepal_length	sepal_width	petal_length	petal_width
count	43.000000	43.000000	43.000000	43.000000
mean	5.023256	3.441860	1.490698	0.234884
std	0.327935	0.352699	0.142785	0.089665
min	4.400000	2.900000	1.300000	0.100000
25%	4.800000	3.150000	1.400000	0.200000
50%	5.000000	3.400000	1.500000	0.200000
75%	5.200000	3.700000	1.550000	0.300000
max	5.700000	4.400000	1.900000	0.400000

In [44]:

```
sep_s_area = 5.023*3.45
pet_s_area = 1.48*0.237
set_t_area = sep_s_area+pet_s_area
set_t_area
```

Out[44]:

```
17.680110000000003
```

In [46]:

```
(virg['sepal_length']<=5.5).value_counts()
```

Out[46]:

```
False    49
True      1
Name: sepal_length, dtype: int64
```

In [48]:

```
virg.drop(virg.loc[virg['sepal_length']<=5.5].index,inplace = True)
virg.shape
```

Out[48]:

```
(49, 5)
```

In [49]:

```
virg.describe()
```

Out[49]:

	sepal_length	sepal_width	petal_length	petal_width
count	49.000000	49.000000	49.000000	49.000000
mean	6.622449	2.983673	5.573469	2.032653
std	0.593459	0.318425	0.536103	0.273395
min	5.600000	2.200000	4.800000	1.400000
25%	6.300000	2.800000	5.100000	1.800000
50%	6.500000	3.000000	5.600000	2.000000
75%	6.900000	3.200000	5.900000	2.300000
max	7.900000	3.800000	6.900000	2.500000

In [50]:

```
sep_v_area = 6.62*2.98
pet_v_area = 5.57*2.032
vir_t_area = sep_v_area+pet_v_area
vir_t_area
```

Out[50]:

```
31.04584
```

In [51]:

```
versi.describe()
```

Out[51]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

In [52]:

```
sep_ve_area = 5.93*2.77
pet_ve_area = 4.26*1.326
set_t_area = sep_ve_area+pet_ve_area
set_t_area
```

Out[52]:

```
22.074859999999997
```

In [54]:

```
dd = setos.append(versi)
df2 = dd.append(virg)
df2
```

Out[54]:

	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
...	...	...	...	...	...
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

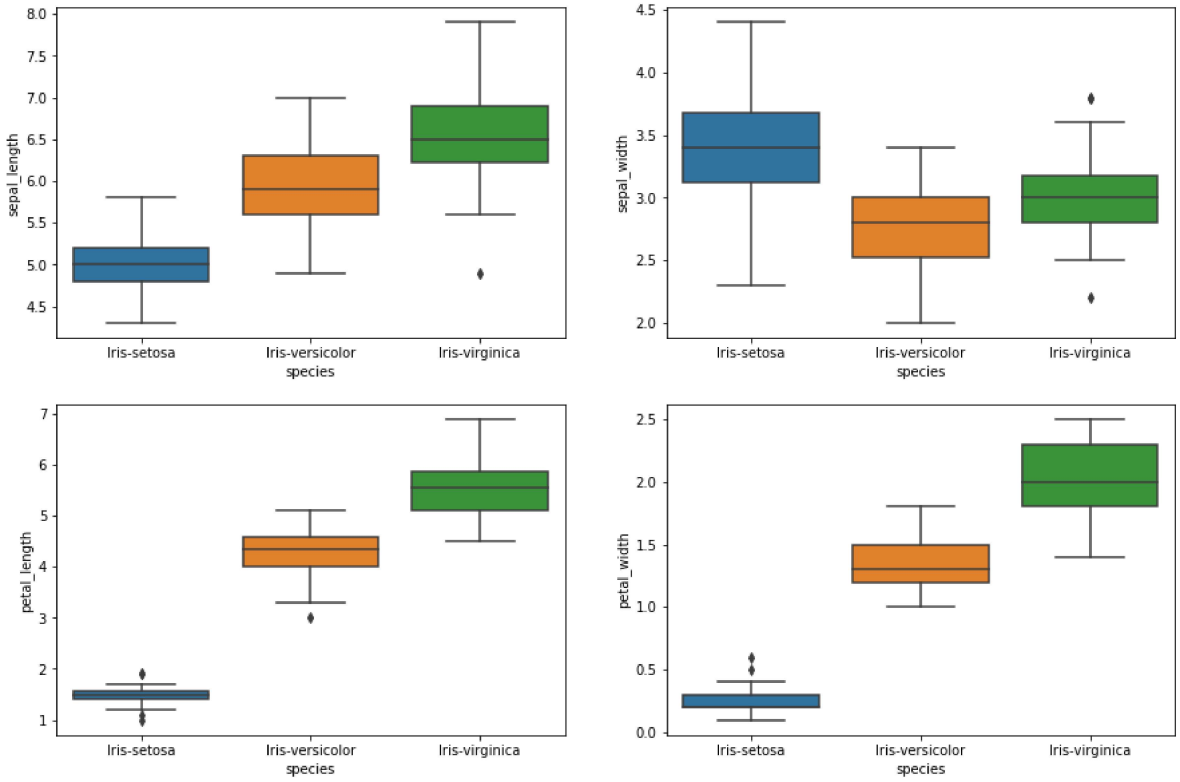
142 rows × 5 columns

In [55]:

```
plt.figure(figsize=(15,10))
plt.subplot(221)
sns.boxplot(x=df['species'],y=df['sepal_length'])
plt.subplot(222)
sns.boxplot(x=df['species'],y=df['sepal_width'])
plt.subplot(223)
sns.boxplot(x=df['species'],y=df['petal_length'])
plt.subplot(224)
sns.boxplot(x=df['species'],y=df['petal_width'])
```

Out[55]:

```
<AxesSubplot:xlabel='species', ylabel='petal_width'>
```



In [56]:

```
df2.columns
```

Out[56]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [57]:

```
x = df2[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']]
y = df2['species']
y
```

Out[57]:

```
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: species, Length: 142, dtype: object
```

In [1]:

```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state = 0)
```

In [ ]:

```

```