```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
dat = pd.read_csv("/content/IRIS.csv")
dat
```

	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

```
dat.head()
```

	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

```
dat.head(2)
```

	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

```
dat.shape
```

```
(150, 5)
```

```
dat.info()
```

```
<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
```

```
dat.isnull()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

```
dat.isnull().sum()
```

	0
sepal_length	0
sepal_width	0
petal_length	0
petal_width	0
species	0

dtype: int64

```
dat.isnull().sum() / dat.shape[0] * 100
```

	0
sepal_length	0.0
sepal_width	0.0
petal_length	0.0
petal_width	0.0
species	0.0

dtype: float64

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
```

```
dat["species"] = le.fit_transform(dat["species"])
```

```
dat.info()
```

```
<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    int64  
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

```
dat.select_dtypes("int")
```

	species
0	0
1	0
2	0
3	0
4	0
...	...
145	2
146	2
147	2
148	2
149	2

150 rows × 1 columns

```
from sklearn.preprocessing import OneHotEncoder

ohe = OneHotEncoder()
```

```
ohe.fit_transform(dat[["species"]])
```

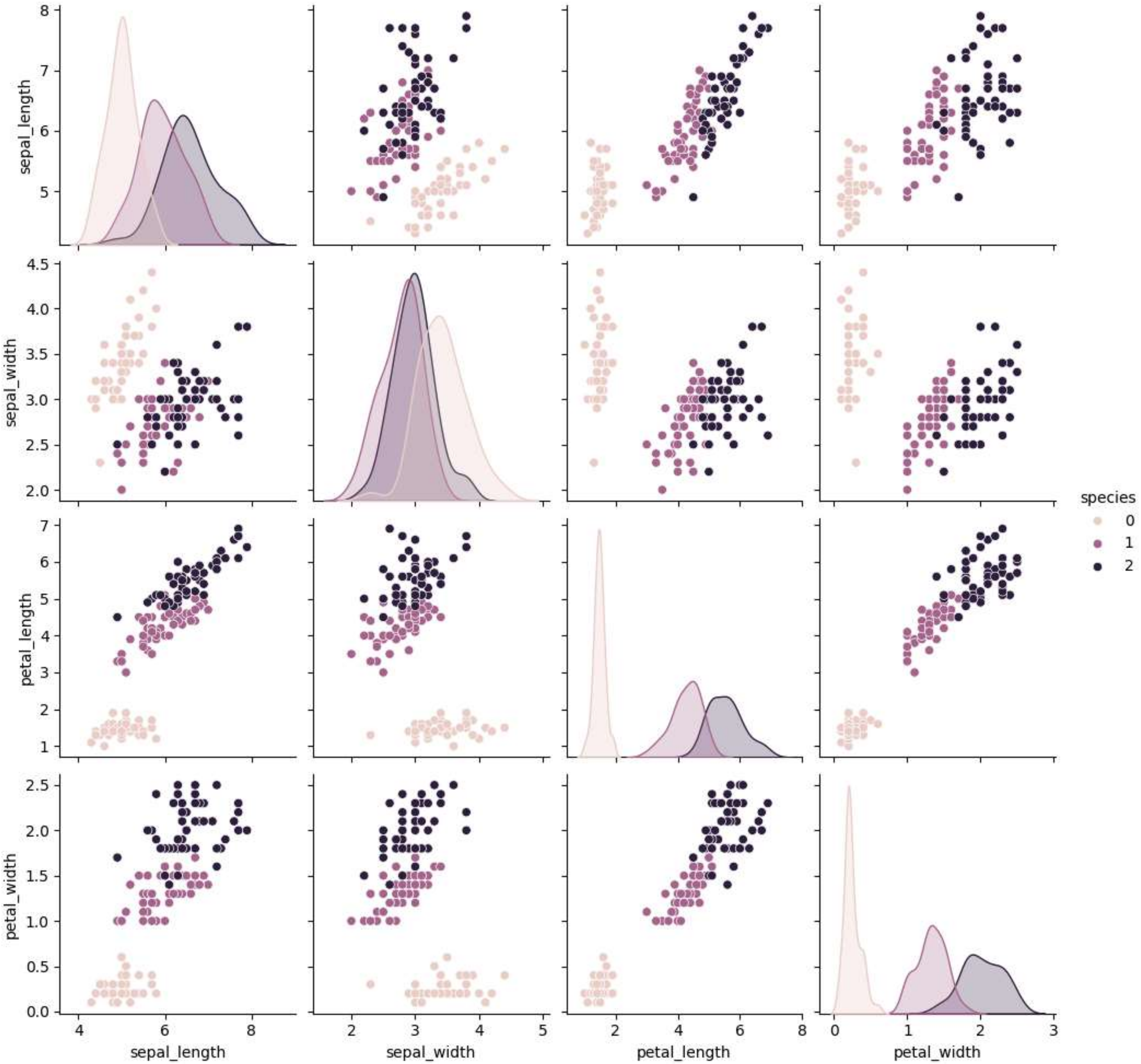
```
<150x3 sparse matrix of type '<class 'numpy.float64'>'
	with 150 stored elements in Compressed Sparse Row format>
```

```
ohe.fit_transform(dat[["species"]]).toarray()
```

```
array([[1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [1., 0., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
           [0., 1., 0.],
```

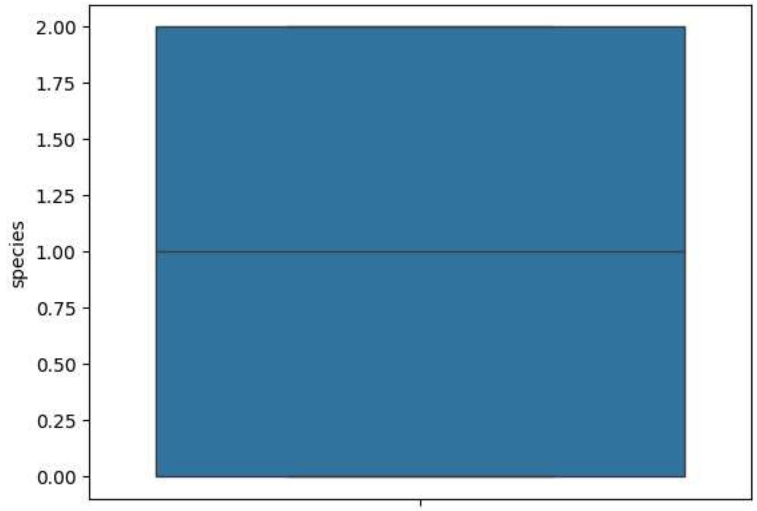
```
import seaborn as sns
sns.pairplot(dat,hue="species")
```

```
<seaborn.axisgrid.PairGrid at 0x7cdafa794410>
```



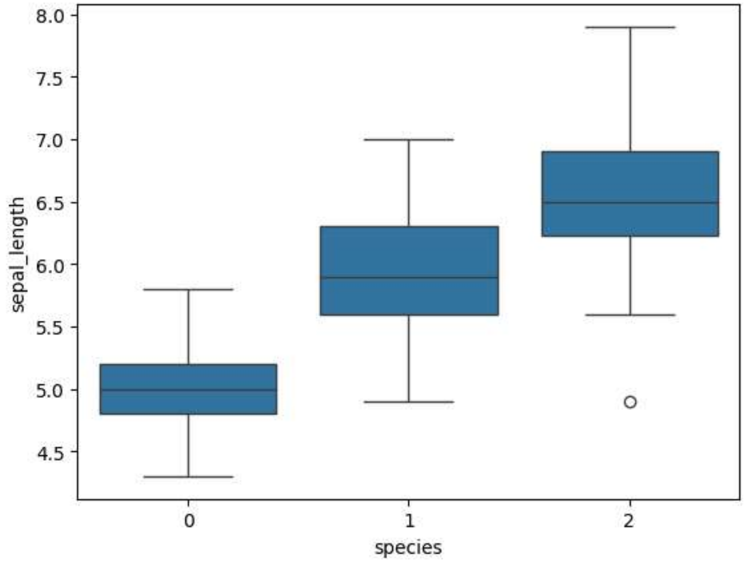
```
sns.boxplot(dat["species"])
```

```
<Axes: ylabel='species'>
```



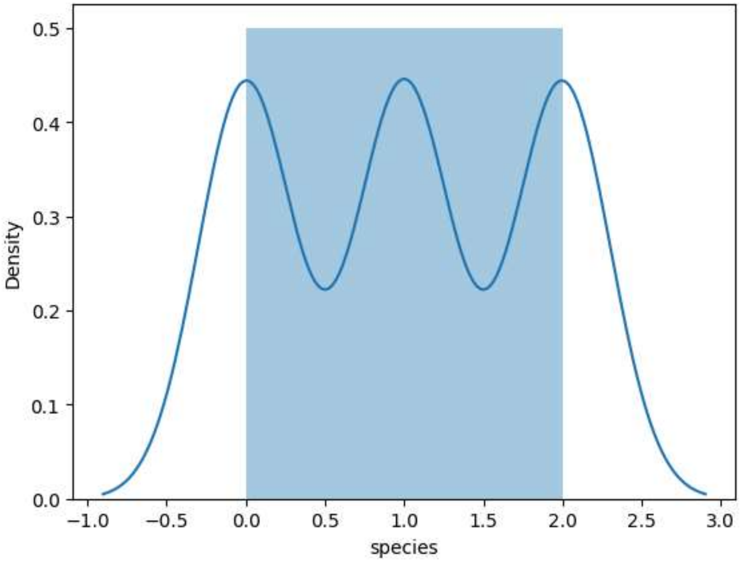
```
sns.boxplot(x="species",y="sepal_length",data=dat)
```

```
<Axes: xlabel='species', ylabel='sepal_length'>
```



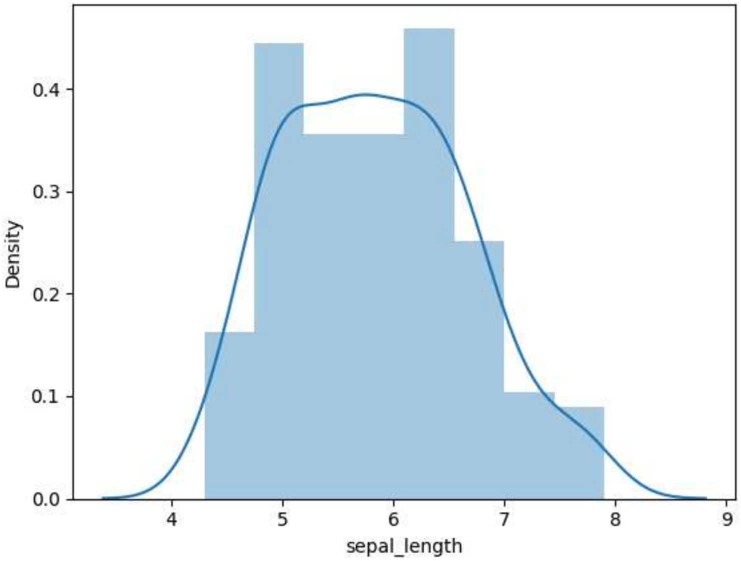
```
sns.distplot(dat["species"])
```

```
<ipython-input-24-3d9f568be3fa>:1: 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(dat["species"])
<Axes: xlabel='species', ylabel='Density'>
```



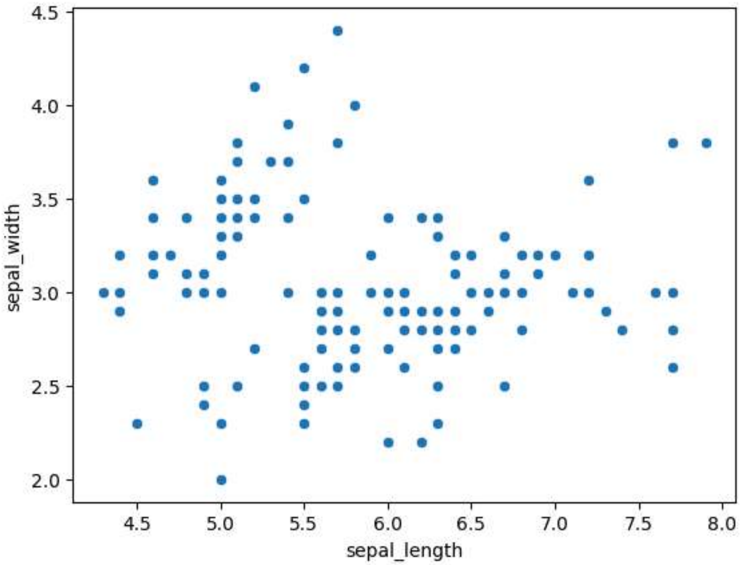
```
sns.distplot(dat["sepal_length"])
```

```
<ipython-input-25-dff069283e45>:1: 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(dat["sepal_length"])
<Axes: xlabel='sepal_length', ylabel='Density'>
```

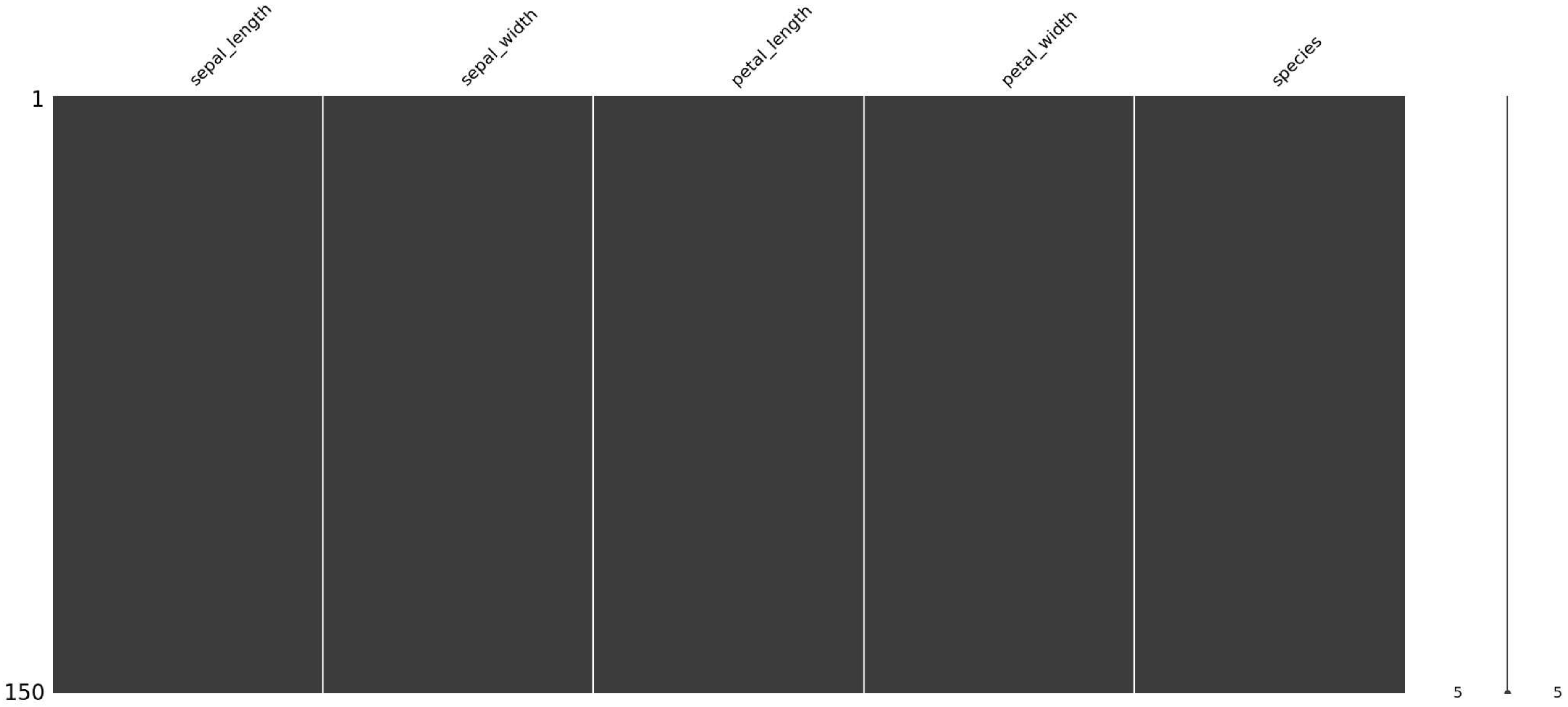


```
sns.scatterplot(x="sepal_length",y="sepal_width",data=dat)
```

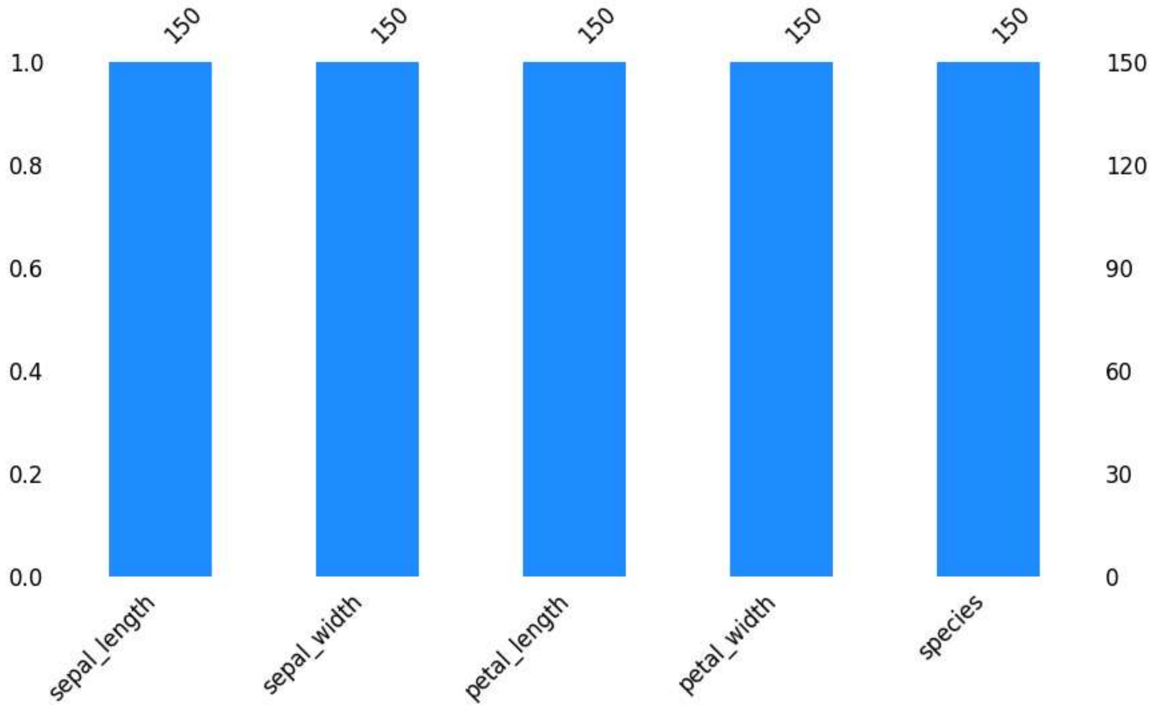
```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```



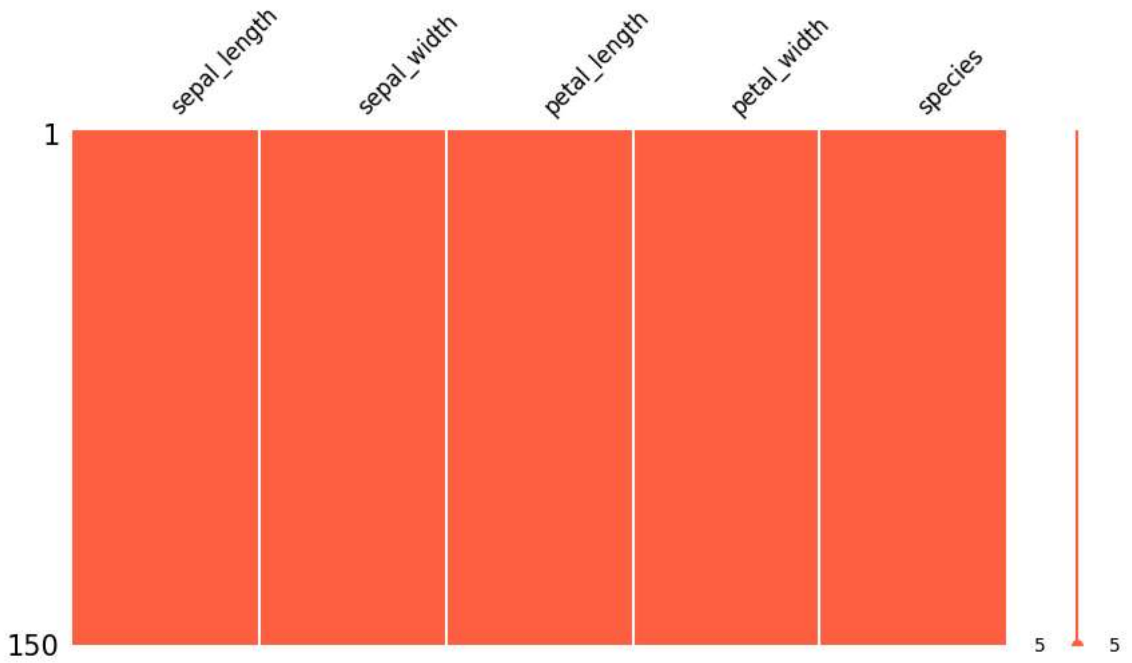
```
dat['sepal_length'] = dat['sepal_length'].astype(int)
import missingno as msno
msno.matrix(dat)
plt.show()
```



```
import missingno
missingno.bar(dat, color="dodgerblue", sort="ascending", figsize=(10,5), fontsize=12);
```

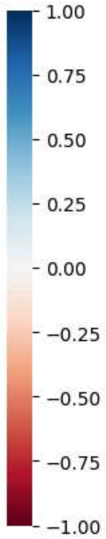


```
missingno.matrix(dat, figsize=(10,5), fontsize=12, color=(1, 0.38, 0.27));
```

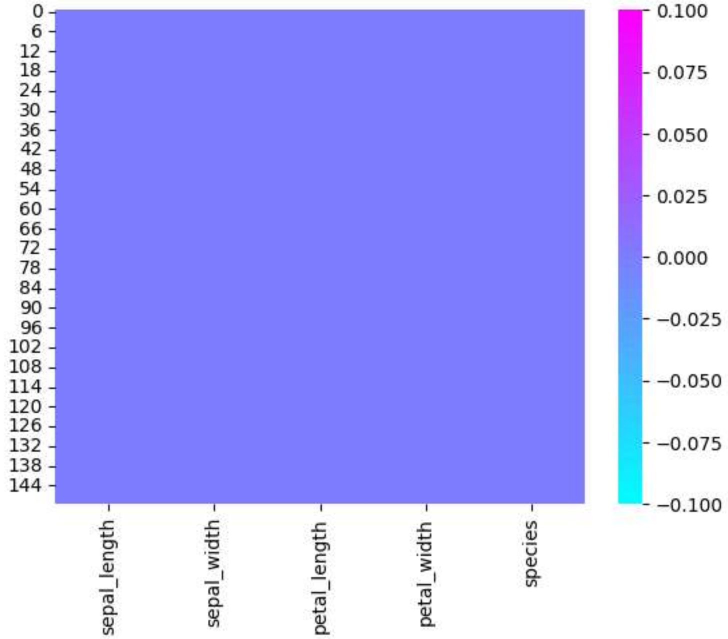


```
missingno.heatmap(dat, figsize=(10,5), fontsize=12);
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/matrix.py:309: UserWarning: Attempting to set identical low and high xlims makes transformation singular; automatically expanding.
  ax.set(xlim=(0, self.data.shape[1]), ylim=(0, self.data.shape[0]))
/usr/local/lib/python3.11/dist-packages/seaborn/matrix.py:309: UserWarning: Attempting to set identical low and high ylims makes transformation singular; automatically expanding.
  ax.set(xlim=(0, self.data.shape[1]), ylim=(0, self.data.shape[0]))
```



```
sns.heatmap(dat.isnull(),cmap='cool');
```



```
X = dat.iloc[ :, :-1]
Y = dat.iloc[ :, -1]
```

```
Y.head(2)
```

	species
0	0
1	0

dtype: int64

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
```

```
X_train.head()
```

	sepal_length	sepal_width	petal_length	petal_width
22	4	3.6	1.0	0.2
15	5	4.4	1.5	0.4
65	6	3.1	4.4	1.4
11	4	3.4	1.6	0.2
42	4	3.2	1.3	0.2

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, Y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
model.coef_
```

```
array([-0.04050024, -0.11332685,  0.19866905,  0.59253822])
```

```
model.intercept_
```

```
0.11463700625225481
```

```
yp = model.predict(X_test)
yp
```

```
array([ 1.19911074, -0.00300739,  2.27013981,  1.32580571,  1.33748529,
        0.06184336,  1.06899617,  1.89637235,  1.40513451,  1.09200844,
        1.7072782 , -0.04995403, -0.10773078, -0.04141981, -0.0427412 ,
        1.37946261,  1.98751954,  1.05541998,  1.259131  ,  1.9704511 ,
        0.02636823,  1.57170211,  0.08171026,  1.91119728,  1.8570516 ,
        1.92757194,  1.80716768,  2.0439749 ,  0.06855362,  0.03770092])
```

```
Y_test
```

	species
73	1
18	0
118	2
78	1
76	1
31	0
64	1
141	2
68	1
82	1
110	2
12	0
36	0
9	0
19	0
56	1
104	2
69	1
55	1
132	2
29	0
127	2
26	0
128	2
131	2
145	2
108	2
143	2
45	0
30	0

dtype: int64

```
model.score(X_test, Y_test) * 100
```

```
94.69259742687085
```

```
model.score(X_train,Y_train)*100
```

```
92.40786080941925
```

```
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train,Y_train)
```