In [79]:

```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, re
```

In [29]:

```
df = sns.load_dataset('iris')
```

In [30]:

```
df
```

Out[30]:

	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 × 5 columns

In [31]:

```
df.isna().sum()
```

Out[31]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [93]:

```
df['petal_length'].max()
```

Out[93]:

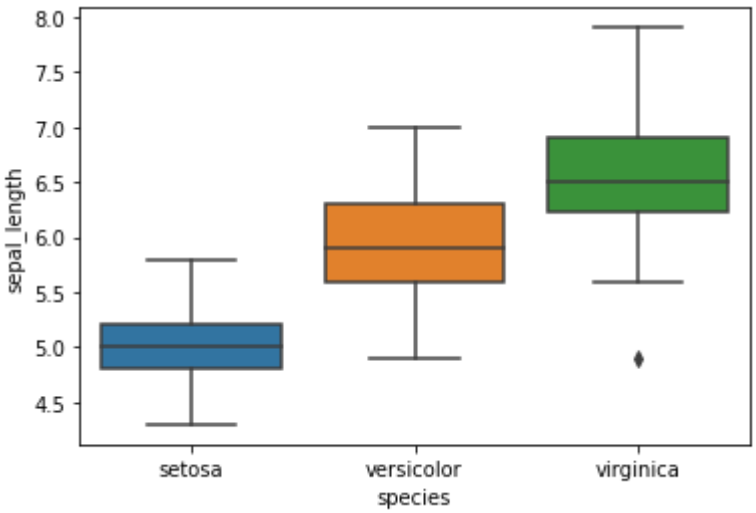
6.9

In [32]:

```
sns.boxplot(x='species', y='sepal_length', data=df)
```

Out[32]:

<AxesSubplot: xlabel='species', ylabel='sepal_length'>

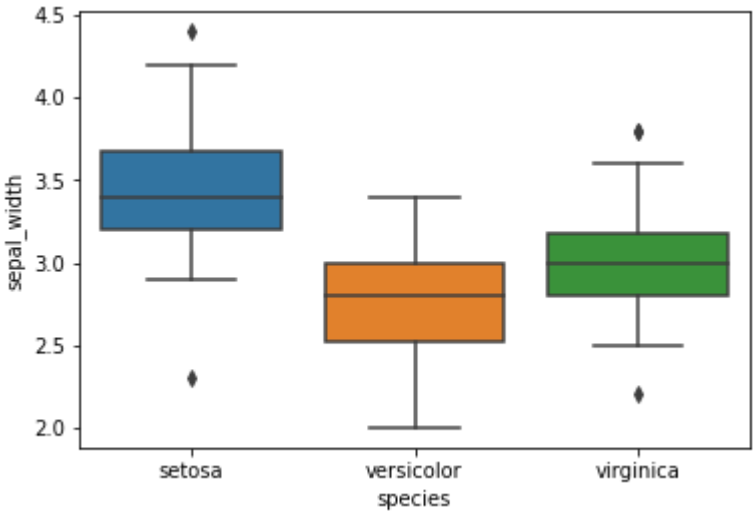


In [33]:

```
sns.boxplot(x='species', y='sepal_width', data=df)
```

Out[33]:

<AxesSubplot: xlabel='species', ylabel='sepal_width'>



In [105]:

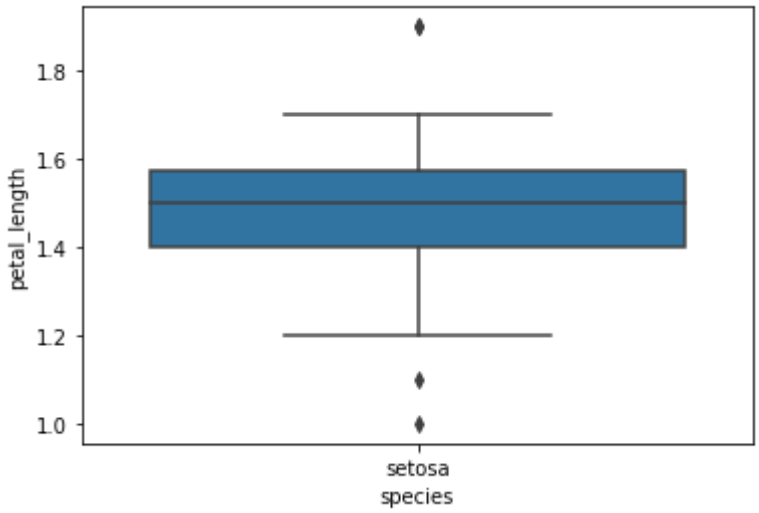
```
setosa = df[df['species'] == 'setosa']
versicolor = df[df['species'] == 'versicolor']
verginica = df[df['species'] == 'virginica']
```

In [107]:

```
sns.boxplot(x='species', y='petal_length', data=setosa)
```

Out[107]:

<AxesSubplot: xlabel='species', ylabel='petal_length'>



In [108]:

```
Q1 = df['petal_length'].quantile(0.25)
Q3 = df['petal_length'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Remove outliers
df = df[(df['petal_length'] >= lower_bound) & (df['petal_length'] <= upper_bound)]
```

In [109]:

```
upper_bound
```

Out[109]:

10.349999999999998

In [110]:

```
df[df['petal_length'] <= upper_bound]
```

Out[110]:

	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

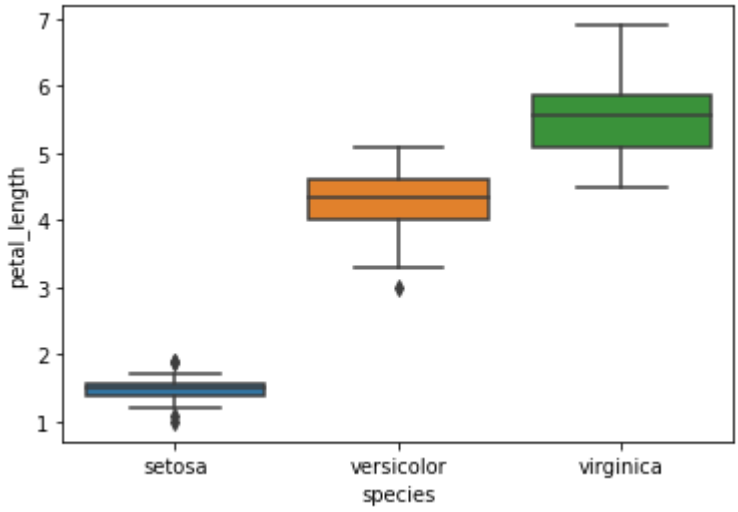
150 rows × 5 columns

In [111]:

```
sns.boxplot(x='species', y='petal_length', data=df)
```

Out[111]:

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



In [112]:

```
# spliting the data set into x, y
```

In [113]:

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

In [114]:

```
x
```

Out[114]:

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
```

In [115]:

```
y
```

Out[115]:

```
array(['setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'virginica', 'virgi
nica',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica', 'virginica', 'virginic
a',
       'virginica', 'virginica', 'virginica'], dtype=object)
```

In [116]:

```
# spliting the dataset into train and test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_s
```

In [117]:

```
naive_bayes_model = GaussianNB()
naive_bayes_model.fit(x_train, y_train)
```

Out[117]:

GaussianNB()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [118]:

```
y_predict = naive_bayes_model.predict(x_test)
```

In [119]:

```
y_predict
```

Out[119]:

```
array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',
       'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolo
r',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',
       'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',
       'setosa'], dtype='<U10')
```

In [120]:

```
y_pred_df = pd.DataFrame(y_predict, columns=['species'])
```

In [121]:

```
y_pred_df
```

Out[121]:

	species
0	virginica
1	versicolor
2	setosa
3	virginica
4	setosa
5	virginica
6	setosa
7	versicolor
8	versicolor
9	versicolor
10	versicolor
11	versicolor
12	versicolor
13	versicolor
14	versicolor
15	setosa
16	versicolor
17	versicolor
18	setosa
19	setosa
20	virginica
21	versicolor
22	setosa
23	setosa
24	virginica
25	setosa
26	setosa
27	versicolor
28	versicolor
29	setosa

In [122]:

```
cm = confusion_matrix(y_test, y_predict)
```

In [123]:

```
TP = cm[1, 1]
FP = cm[0, 1]
TN = cm[0, 0]
FN = cm[1, 0]
```

In [124]:

```
accuracy = (TP + TN) / (TP + TN + FP + FN)
error_rate = (FP + FN) / (TP + TN + FP + FN)
precision = TP / (TP + FP)
recall = TP / (TP + FN)
```

In [125]:

```
cm
```

Out[125]:

```
array([[11,  0,  0],
       [ 0, 13,  0],
       [ 0,  1,  5]])
```

In [126]:

```
accuracy
```

Out[126]:

```
1.0
```

In [127]:

```
error_rate
```

Out[127]:

```
0.0
```

In [128]:

```
precision
```

Out[128]:

```
1.0
```

In [129]:

```
recall
```

Out[129]:

```
1.0
```

In [ ]:

```

```

In [ ]:

```
```