

plotting in python on iris dataset

Written by: M.Danish Azeem Date: 07.12.2023 Email:m danishazeem365@gmail.com

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

# import dataset
phool = sns.load_dataset("iris")
phool
```

	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 x 5 columns]
```

```
phool.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
```

```
phool.isnull().sum()
```

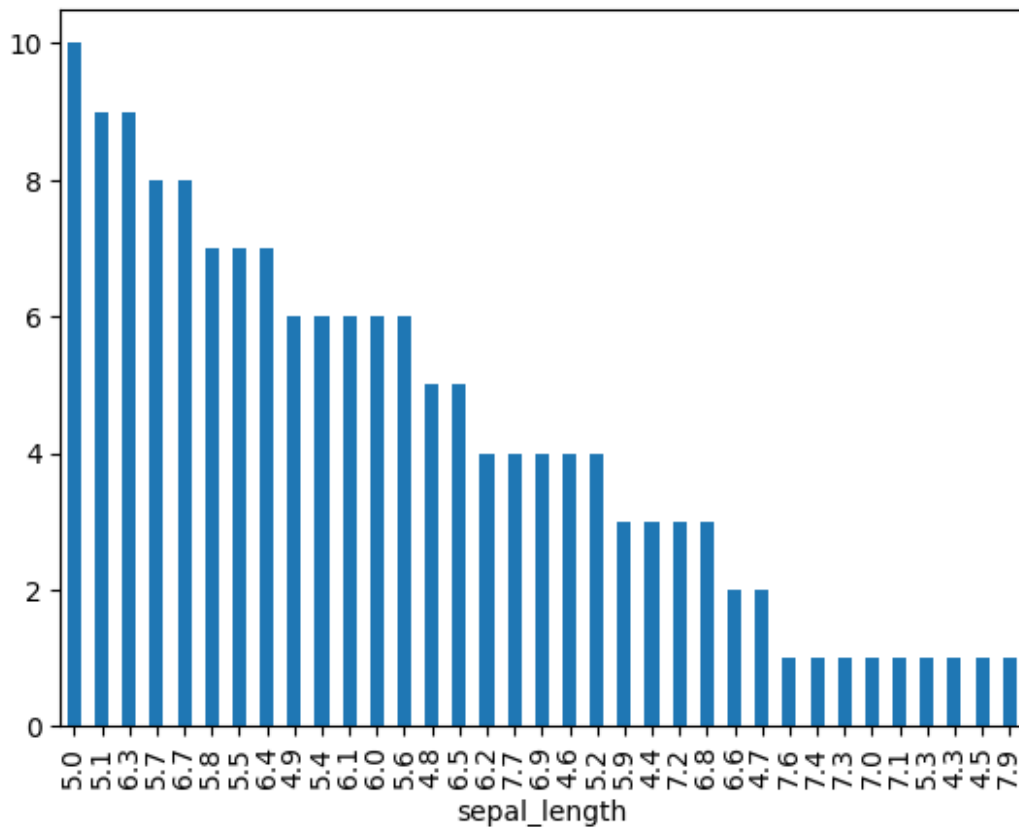
```
sepal_length    0
sepal_width     0
```

```

petal_length    0
petal_width     0
species         0
dtype: int64

phool.sepal_length.value_counts().plot(kind="bar")
<Axes: xlabel='sepal_length'>

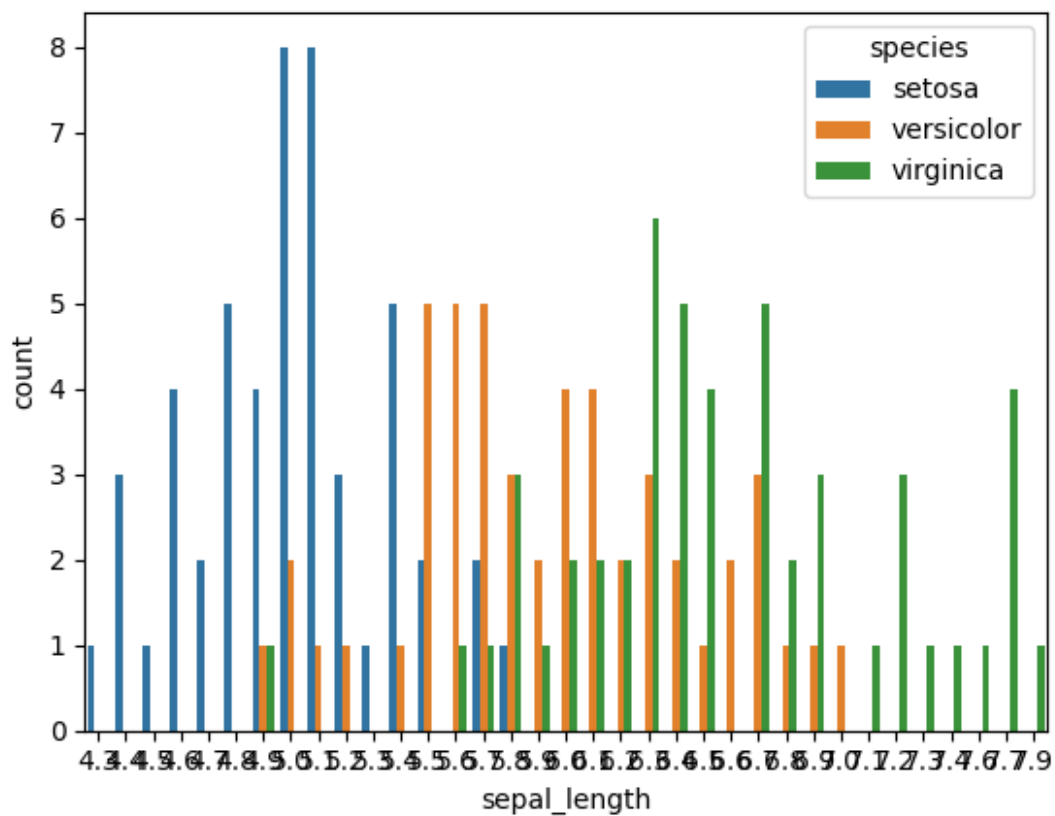
```



```

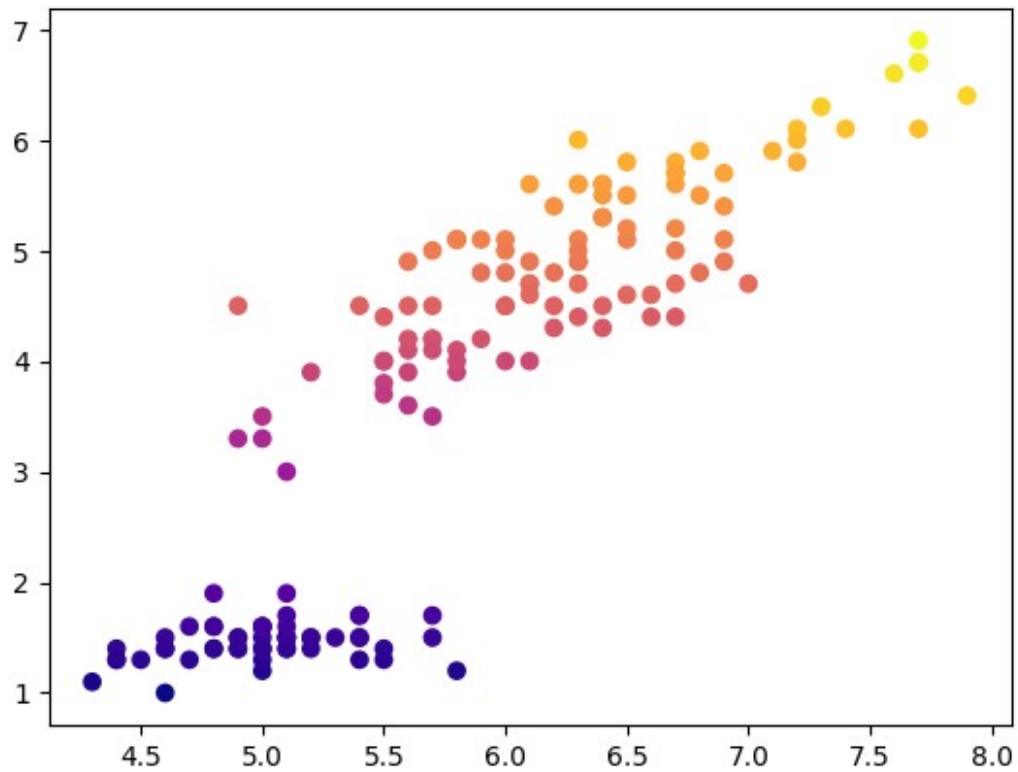
sns.countplot(x="sepal_length", hue="species", data=phool)
<Axes: xlabel='sepal_length', ylabel='count'>

```



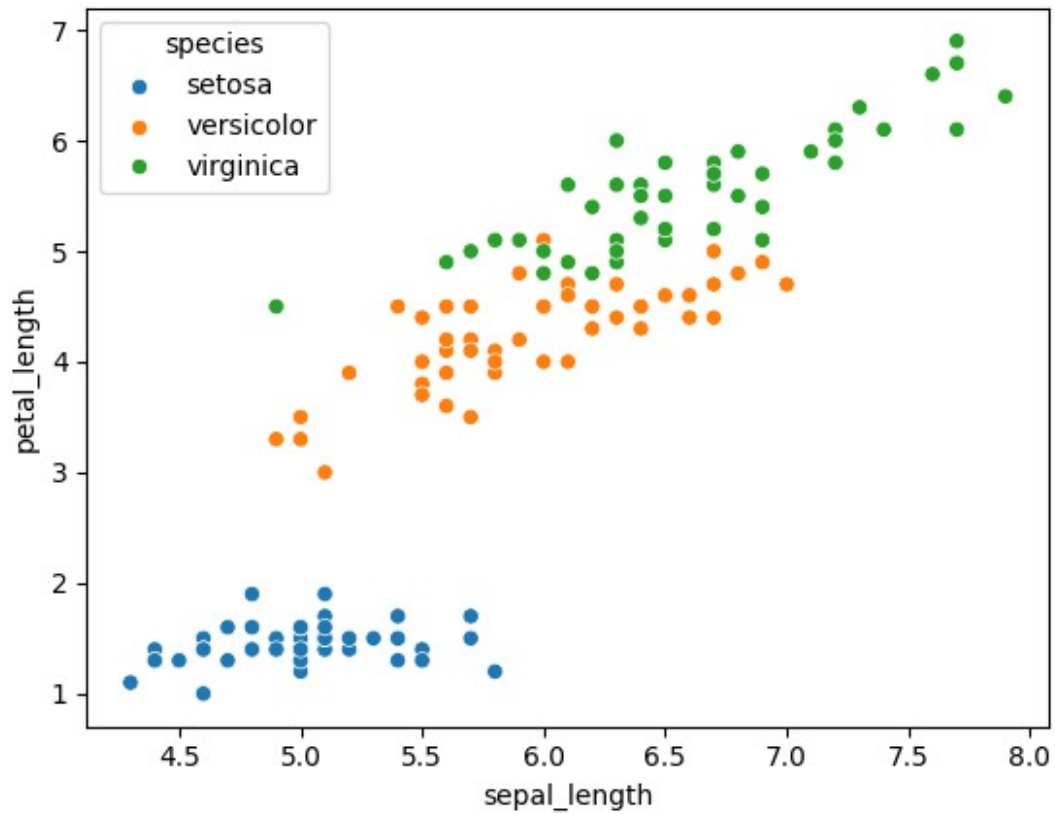
```
plt.scatter(data=phool, x="sepal_length", y="petal_length",
c=phool["petal_length"], cmap="plasma")
```

```
<matplotlib.collections.PathCollection at 0x231ad5e2d50>
```



```
sns.scatterplot(data=phool, x="sepal_length", y="petal_length",  
hue="species")
```

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

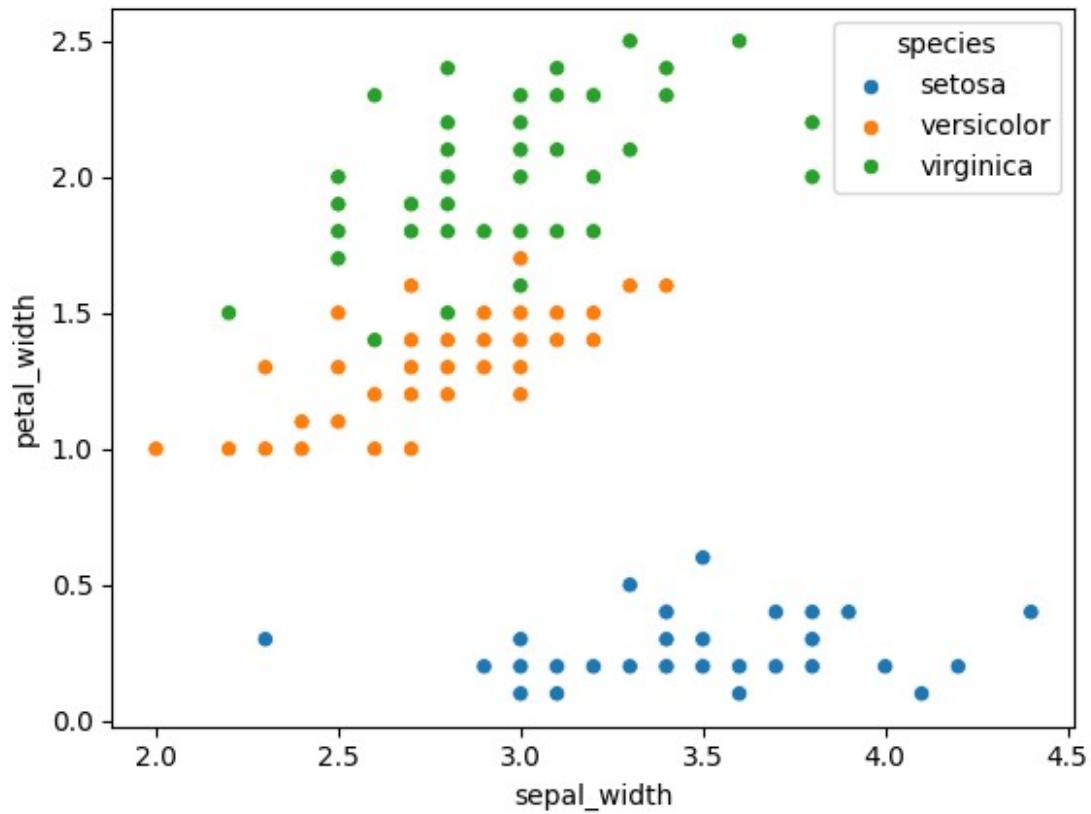


```
phool.columns
```

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

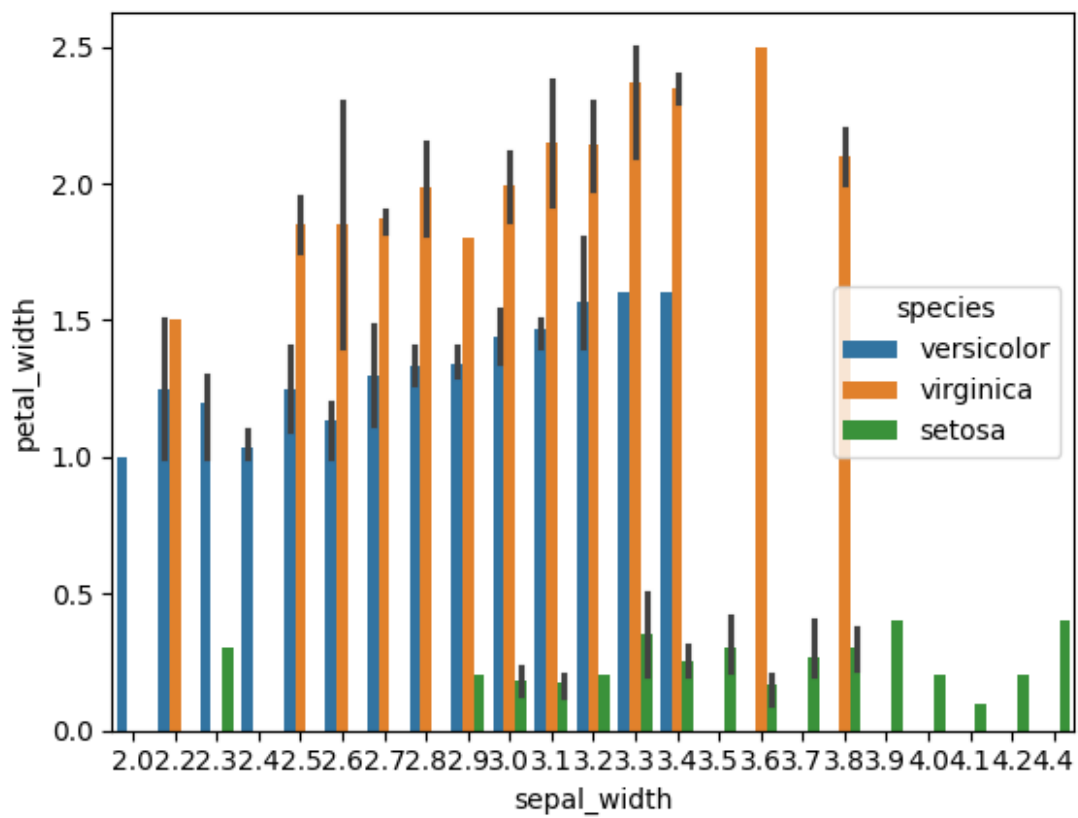
```
sns.scatterplot(data=phool, x="sepal_width", y="petal_width",  
               hue="species")
```

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



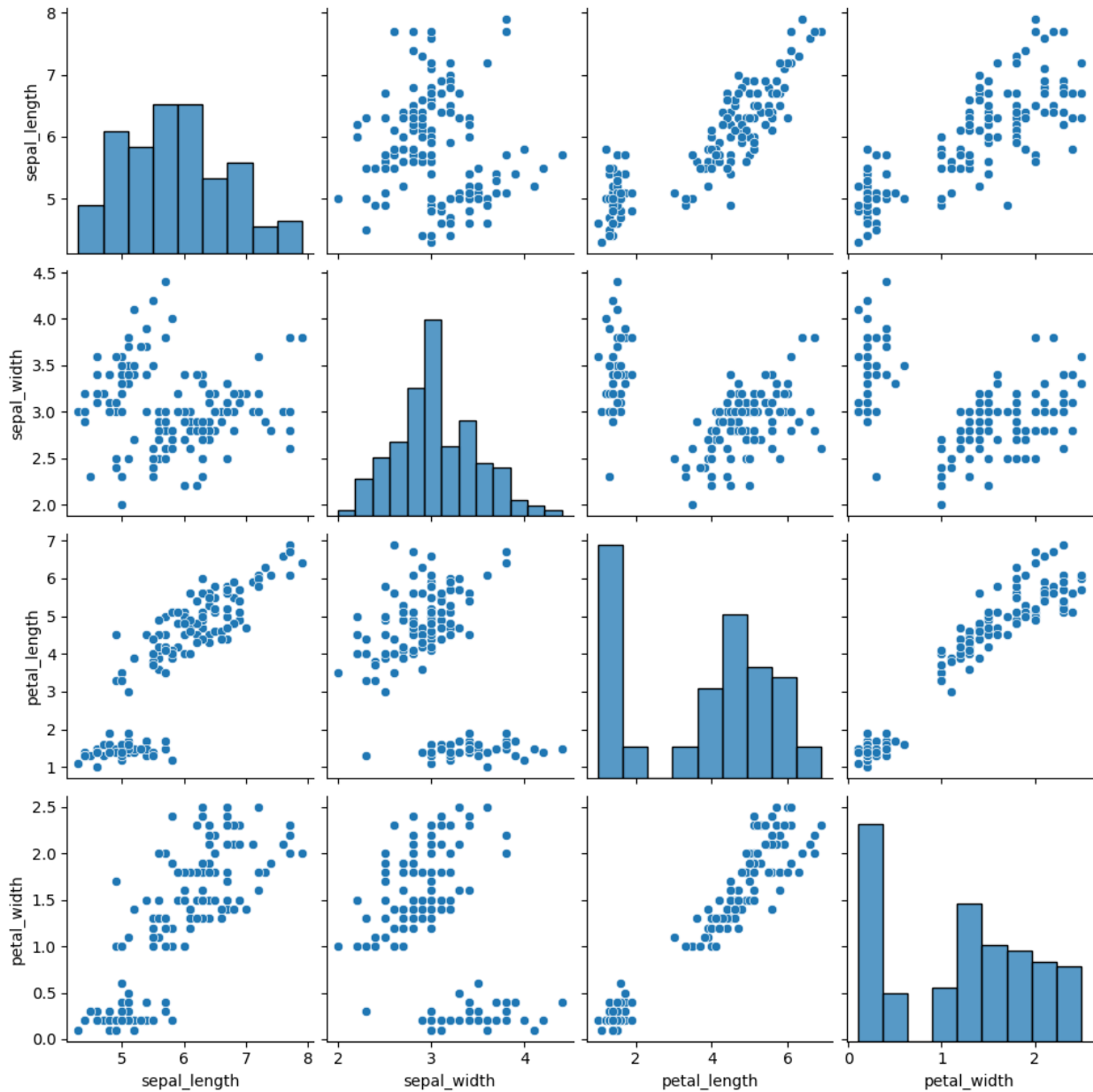
```
sns.barplot(data=phool, x="sepal_width", y="petal_width",  
hue="species")
```

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



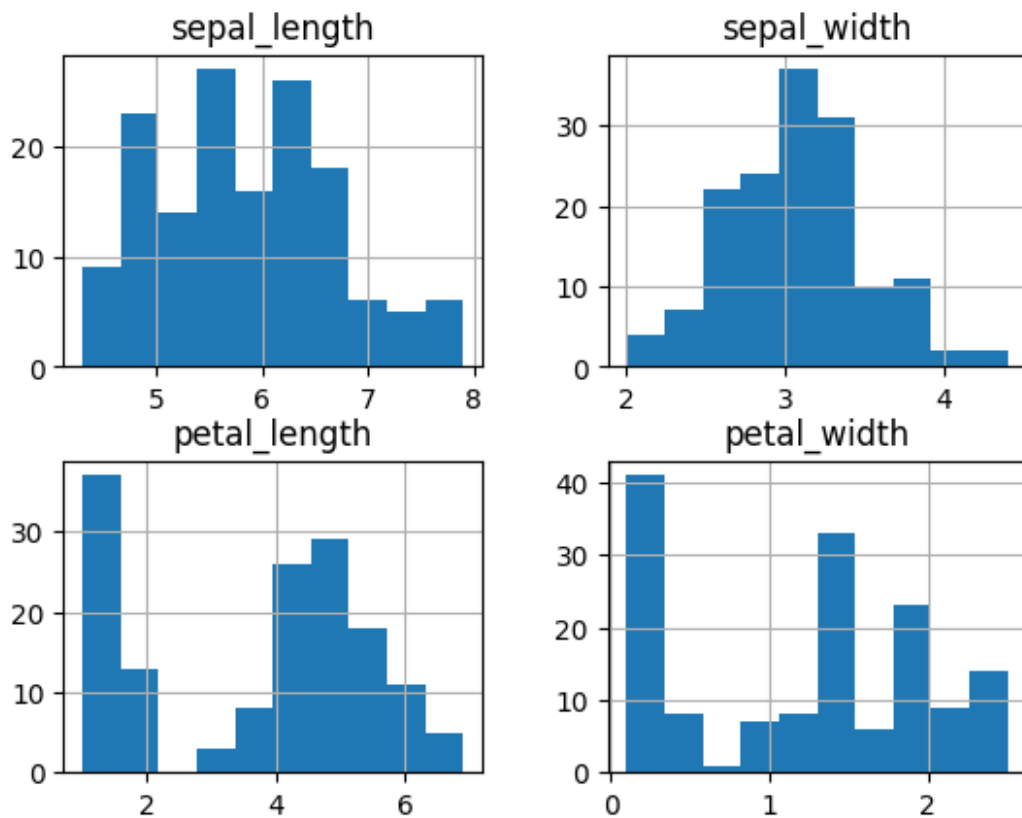
```
sns.pairplot(phool)
```

```
<seaborn.axisgrid.PairGrid at 0x231ad7d3380>
```

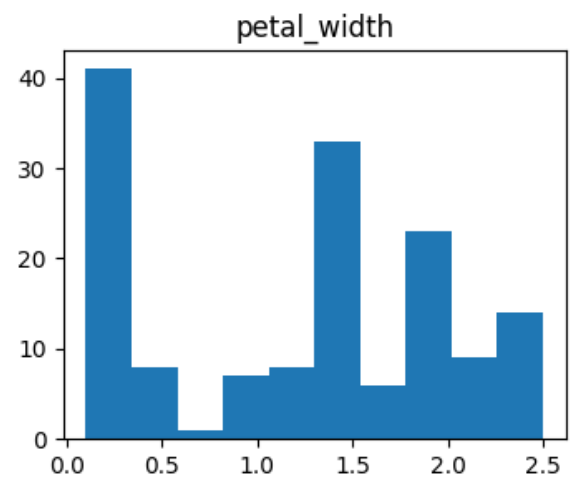
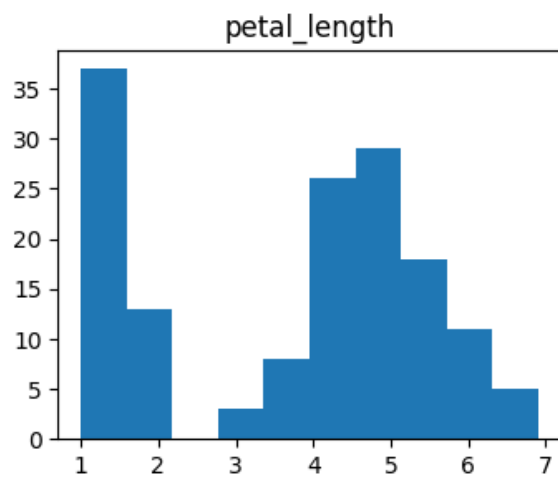
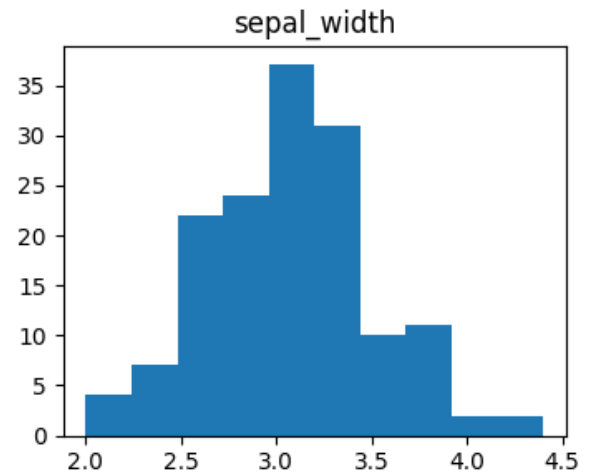
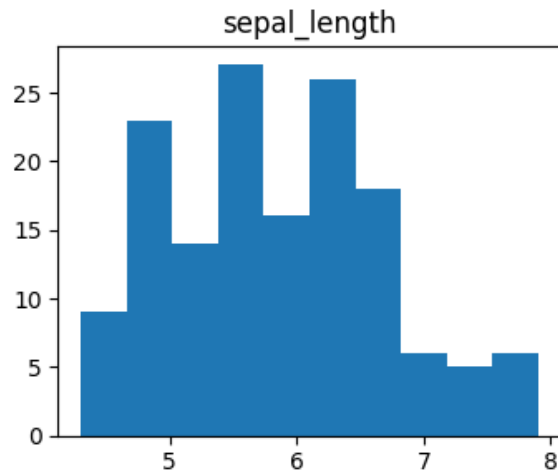


```
phool.hist()
```

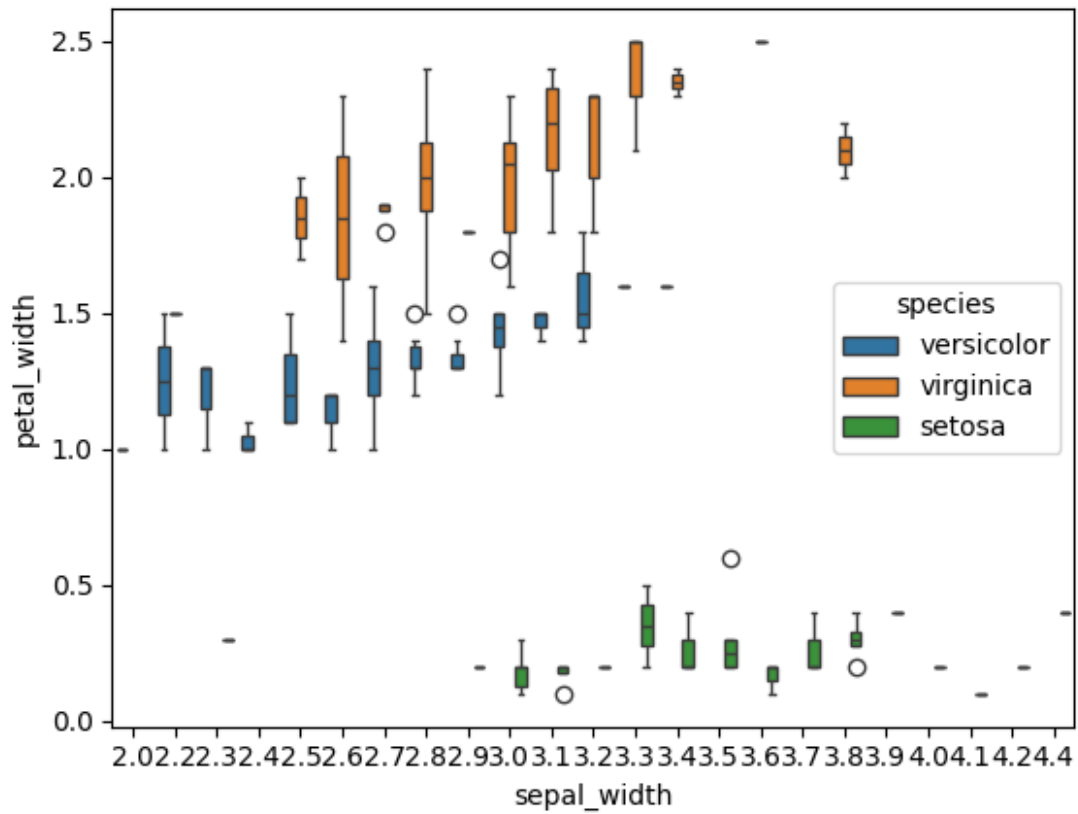
```
array([[<Axes: title={'center': 'sepal_length'}>,
        <Axes: title={'center': 'sepal_width'}>],
       [<Axes: title={'center': 'petal_length'}>,
        <Axes: title={'center': 'petal_width'}>]], dtype=object)
```

```
phool.hist(bins=10, figsize=(9,7), grid=False)
array([[<Axes: title={'center': 'sepal_length'}>,
        <Axes: title={'center': 'sepal_width'}>],
       [<Axes: title={'center': 'petal_length'}>,
        <Axes: title={'center': 'petal_width'}>]], dtype=object)
```

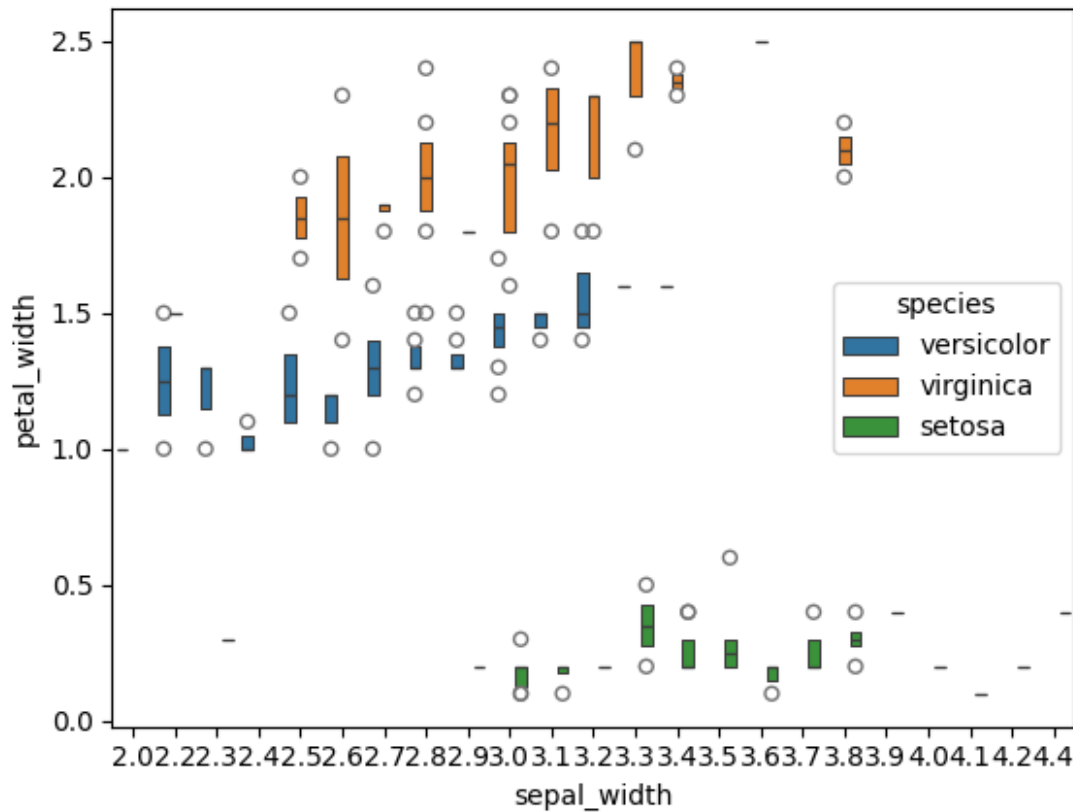


```
sns.boxplot(data=phool, x="sepal_width", y="petal_width",  
hue="species")  
<Axes: xlabel='sepal_width', ylabel='petal_width'>
```



```
sns.boxenplot(data=phool, x="sepal_width", y="petal_width",
hue="species")
```

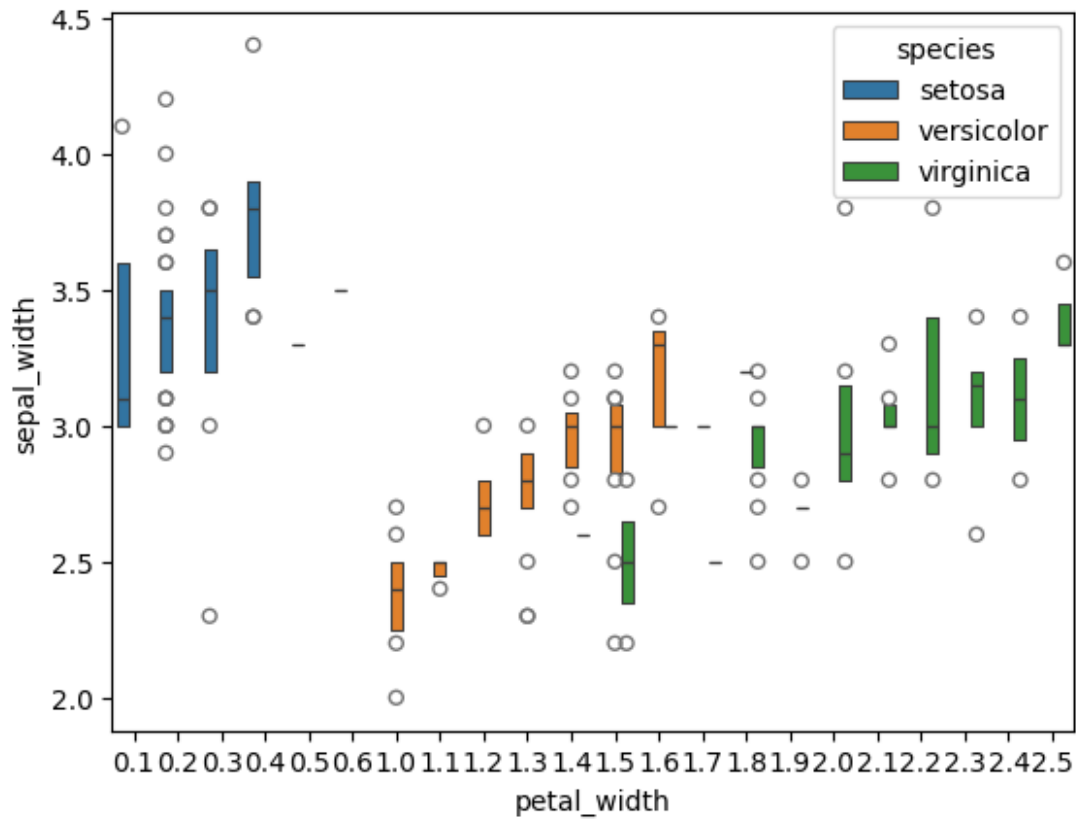
```
<Axes: xlabel='sepal_width', ylabel='petal_width'>
```



Assignment : How to change the x and y axis titles?

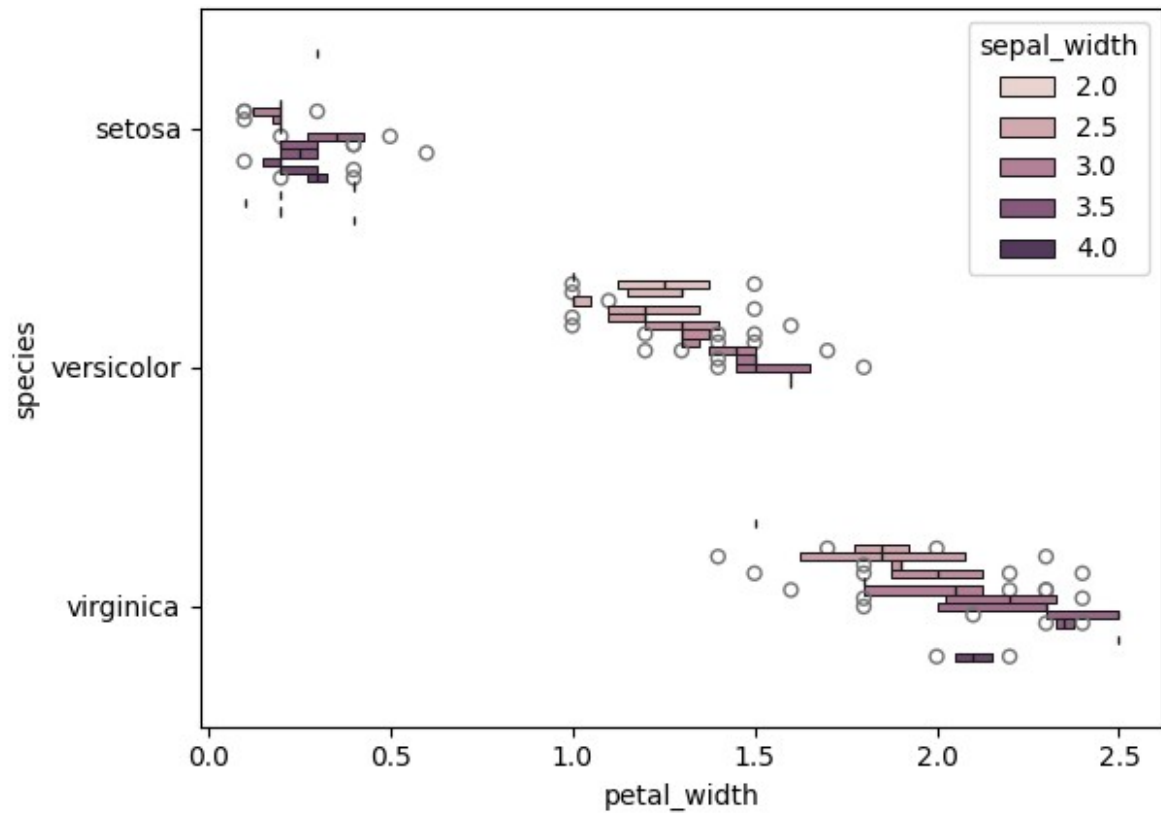
```
sns.boxenplot(data=phool, x="petal_width", y="sepal_width",
hue="species")
```

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



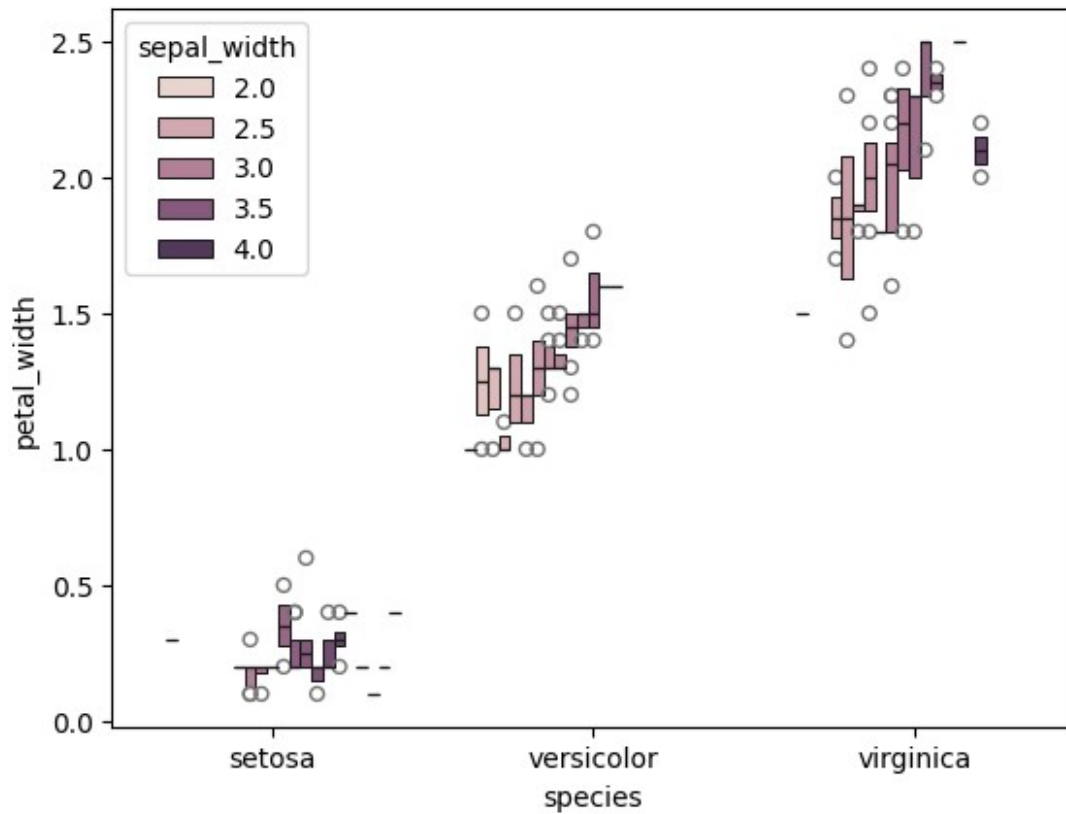
```
sns.boxenplot(data=phool, x="petal_width", y="species",
hue="sepal_width")
```

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



```
sns.boxenplot(data=phool, x="species", y="petal_width",  
hue="sepal_width")
```

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

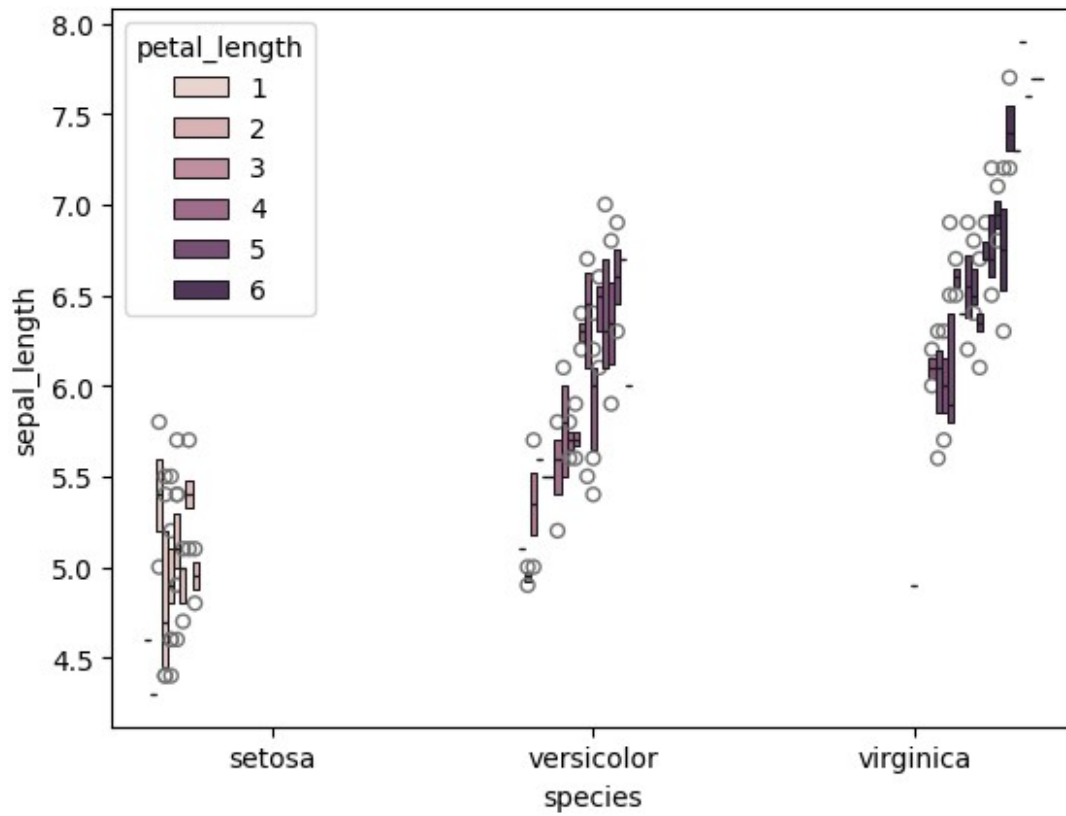


```
phool.columns
```

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

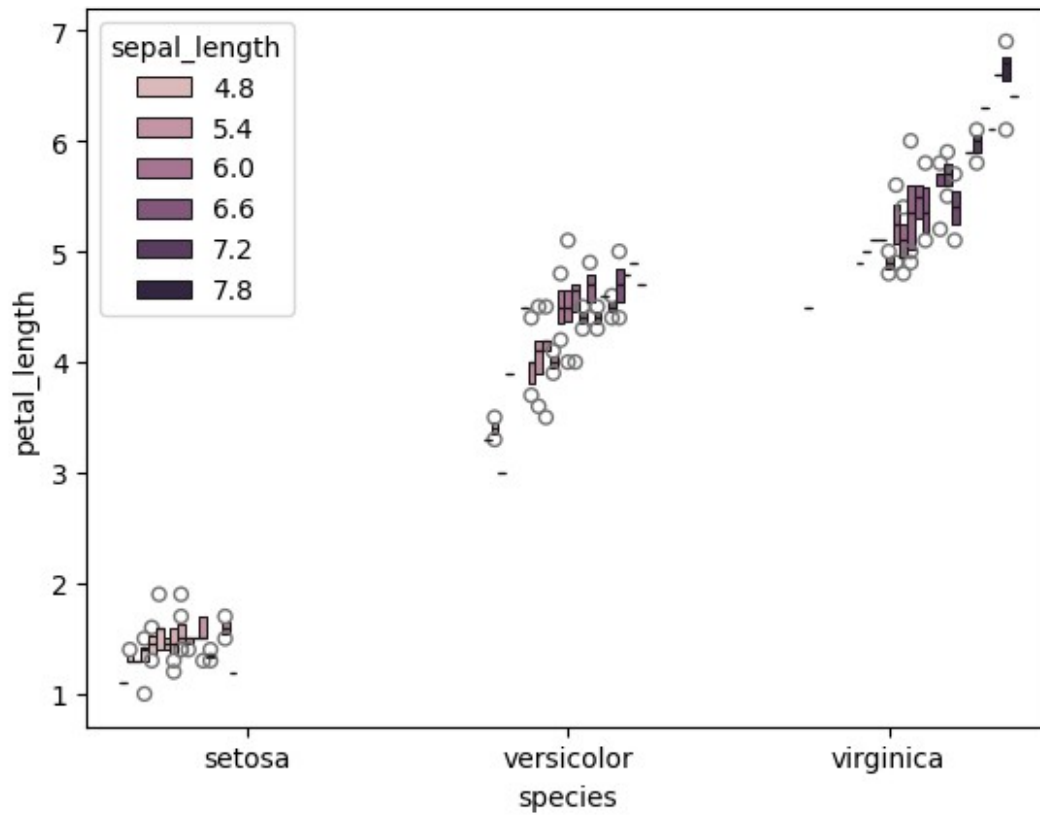
```
sns.boxenplot(data=phool, x="species", y="sepal_length",  
              hue="petal_length")
```

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



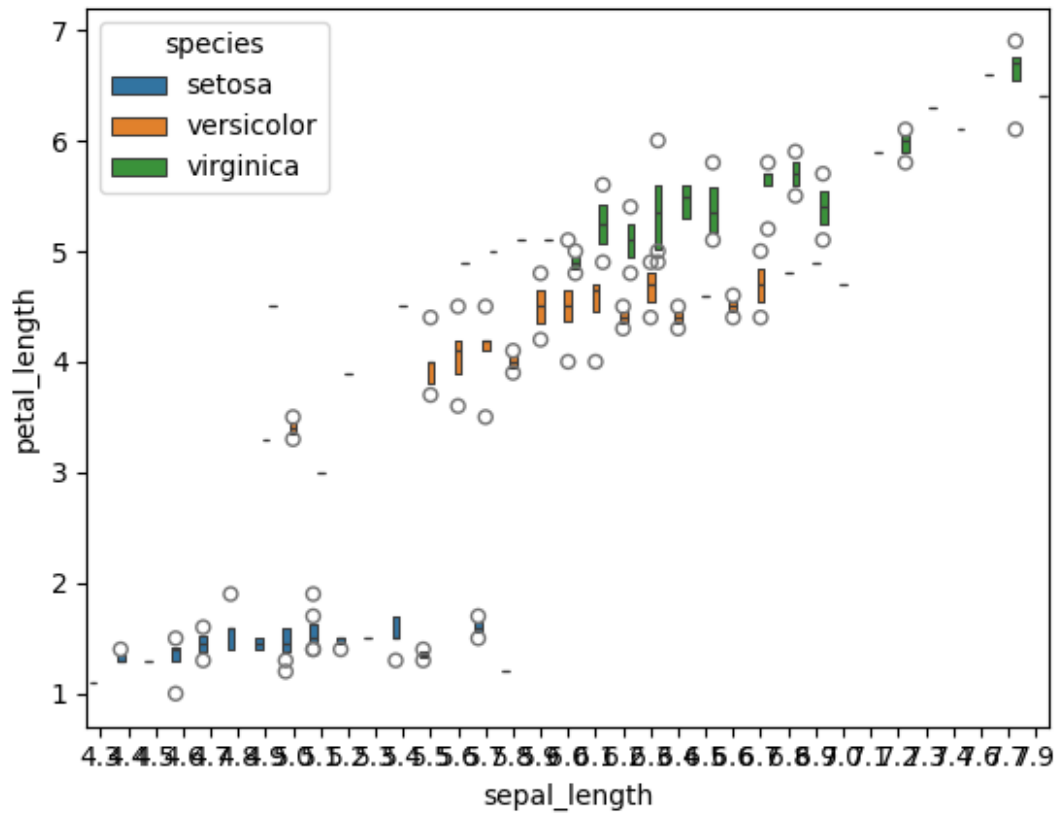
```
sns.boxenplot(data=phool, x="species", y="petal_length",  
hue="sepal_length")
```

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

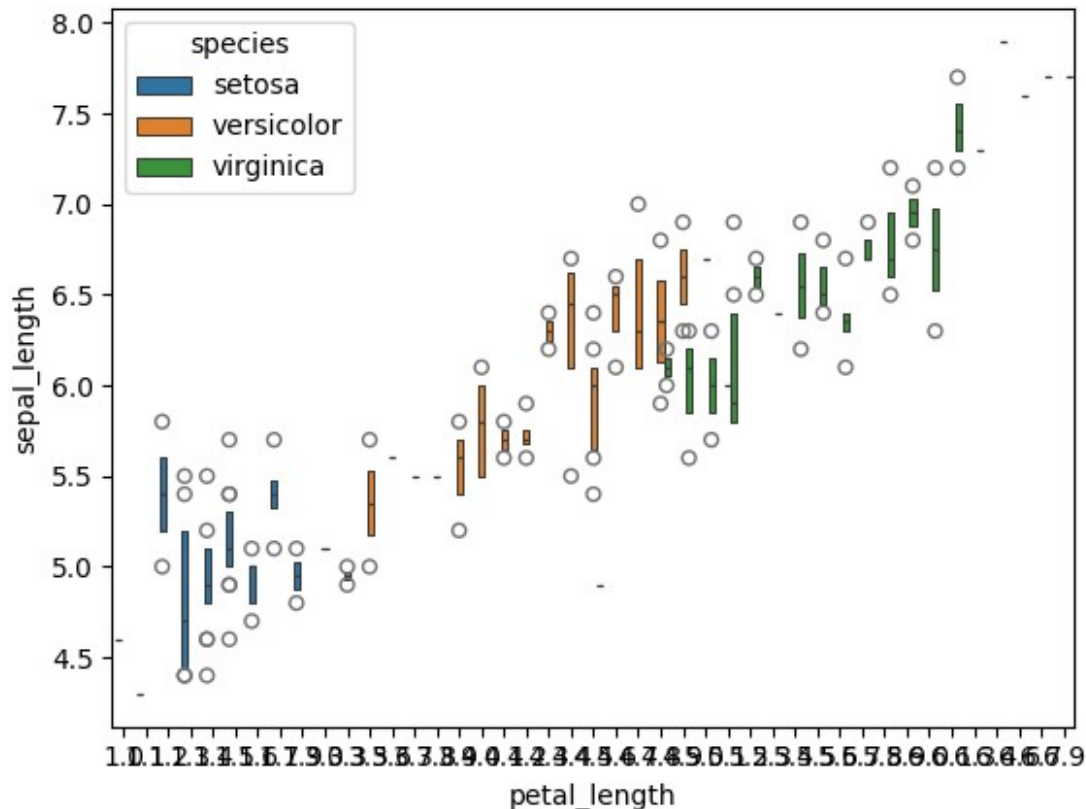
```
sns.boxenplot(data=phool, x="sepal_length", y="petal_length",  
hue="species")
```

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



```
sns.boxenplot(data=phool, x="petal_length", y="sepal_length",
hue="species")
```

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



```
phool.columns
```

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

```
g = sns.FacetGrid(phool, hue="species", col="petal_length",  
margin_titles=True,  
                  palette={"setosa": "seagreen", "versicolor": "gray",  
"virginica": "blue"})  
g = g.map(plt.scatter, "sepal_length", "petal_length",  
edgecolor="b").add_legend()
```

```
# error in this on iris dataset
```

```
# g = sns.FacetGrid(phool, hue="sepal_width", col="sepal_length",  
margin_titles=True,  
#                  palette={1:"seagreen", 0:"gray"})  
# g=g.map(plt.scatter, "fare", "species",edgecolor="b").add_legend()  
  
# Convert sepal_width to a categorical variable
```

```

# phool['sepal_width_category'] = pd.cut(phool['sepal_width'], bins=[-
np.inf, 2, np.inf], labels=['Low', 'High'])

# not work and error

# # Create a FacetGrid using the new categorical column for hue
# g = sns.FacetGrid(phool, hue="sepal_width_category",
col="sepal_length", margin_titles=True,
# palette={"Low": "seagreen", "High": "gray"})

# # Use scatter plot and add legend
# g = g.map(plt.scatter, "fare", "species",
edgecolor="b").add_legend()

# this method work

g = sns.FacetGrid(phool, hue="sepal_width_category",
col="sepal_length", margin_titles=True,
palette={"Low": "seagreen", "High": "gray"})

# Use scatter plot and add legend
g = g.map(plt.scatter, "sepal_width", "species",
edgecolor="b").add_legend()

```

```

# error in this on iris dataset

# g = sns.FacetGrid(phool, hue="species", col="species",
margin_titles=True,
# palette={1:"seagreen", 0:"gray"})
# g=g.map(plt.scatter, "fare",
"sepal_length",edgecolor="b").add_legend()

# not work and error

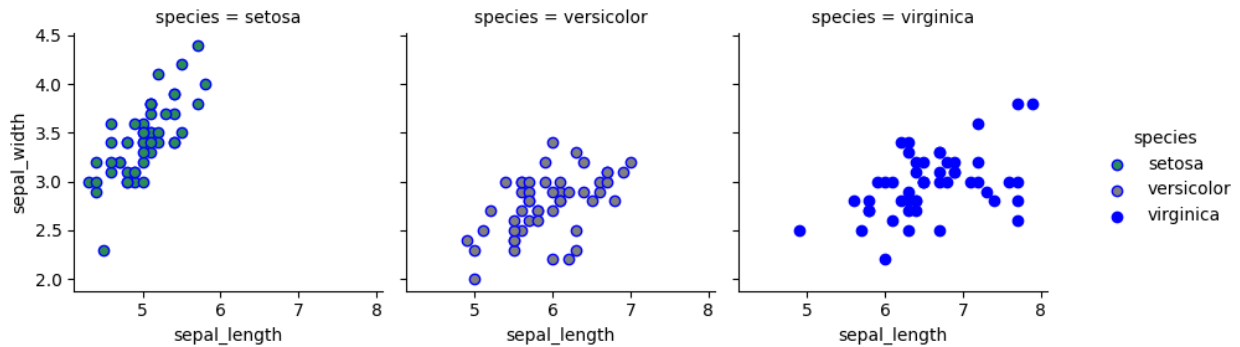
# g = sns.FacetGrid(phool, hue="species", col="species",
margin_titles=True,
# palette={"setosa": "seagreen", "versicolor":
"gray", "virginica": "blue"})
# g = g.map(plt.scatter, "fare", "sepal_length",
edgecolor="b").add_legend()

# this method work

# Assuming you have columns like "sepal_length" and "sepal_width"
g = sns.FacetGrid(phool, hue="species", col="species",

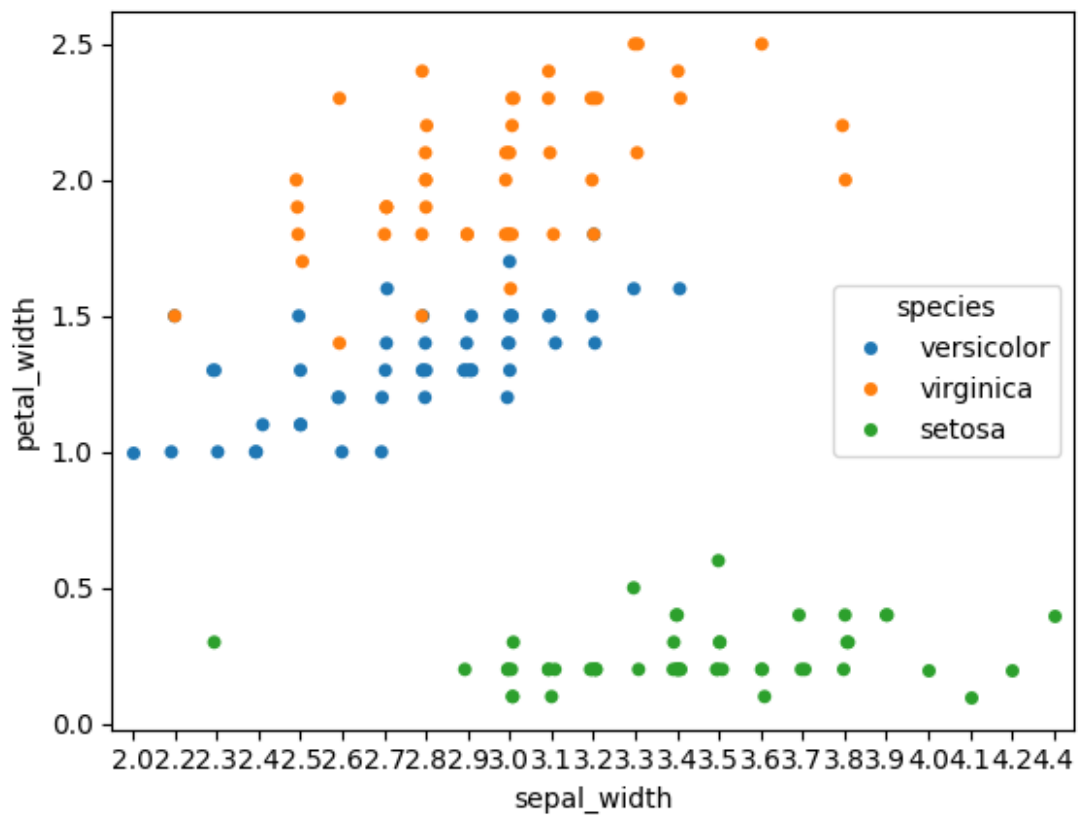
```

```
margin_titles=True,
        palette={"setosa": "seagreen", "versicolor": "gray",
"virginica": "blue"})
g = g.map(plt.scatter, "sepal_length", "sepal_width",
edgecolor="b").add_legend()
```



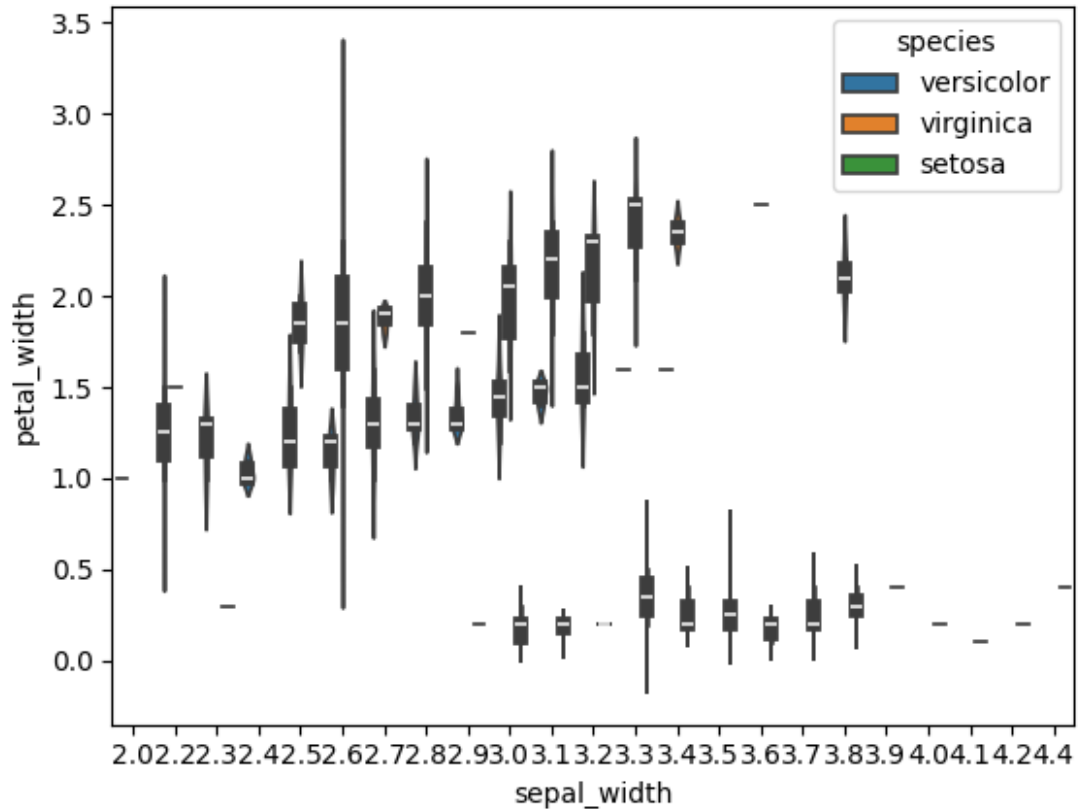
```
sns.stripplot(data=phool, x="sepal_width", y="petal_width",
hue="species")
```

<Axes: xlabel='sepal_width', ylabel='petal_width'>



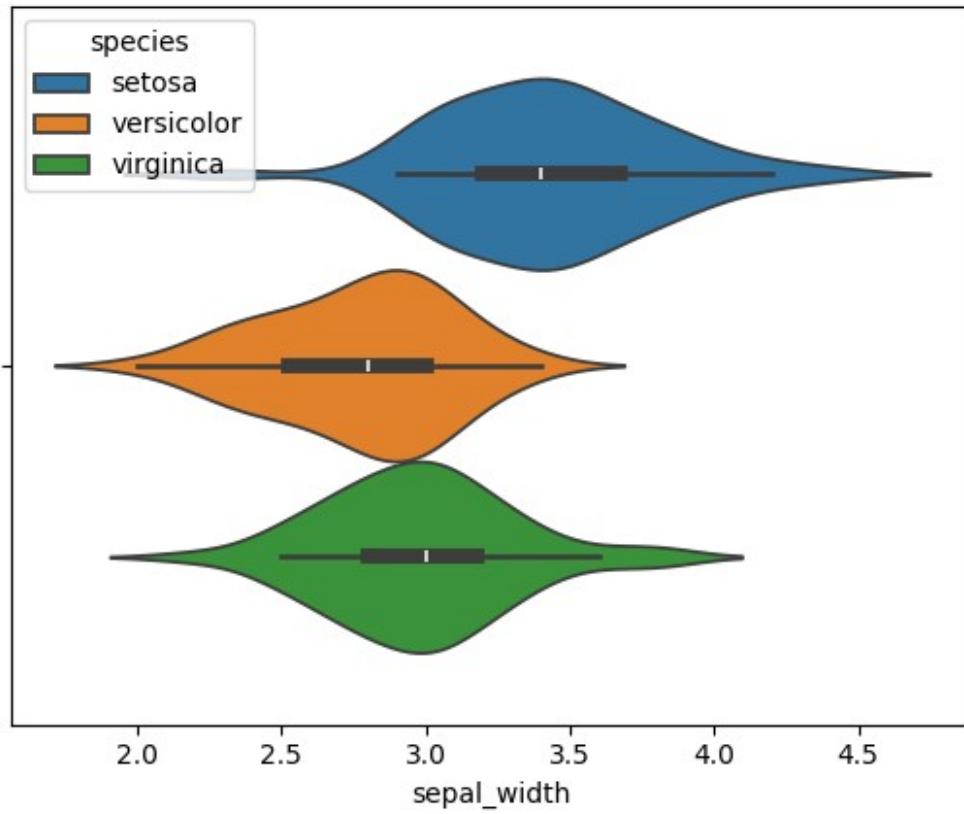
```
sns.violinplot(data=phool, x="sepal_width", y="petal_width",  
hue="species")
```

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

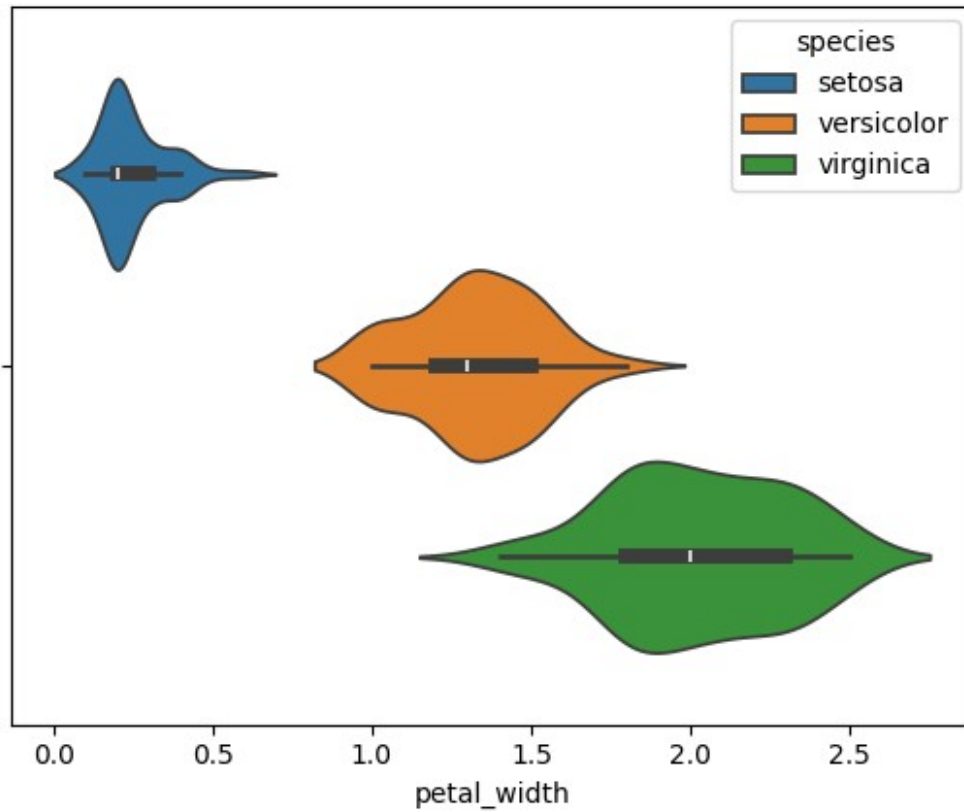


```
sns.violinplot(data=phool, x="sepal_width", hue="species")
```

```
<Axes: xlabel='sepal_width'>
```



```
sns.violinplot(data=phool, x="petal_width", hue="species")  
<Axes: xlabel='petal_width'>
```

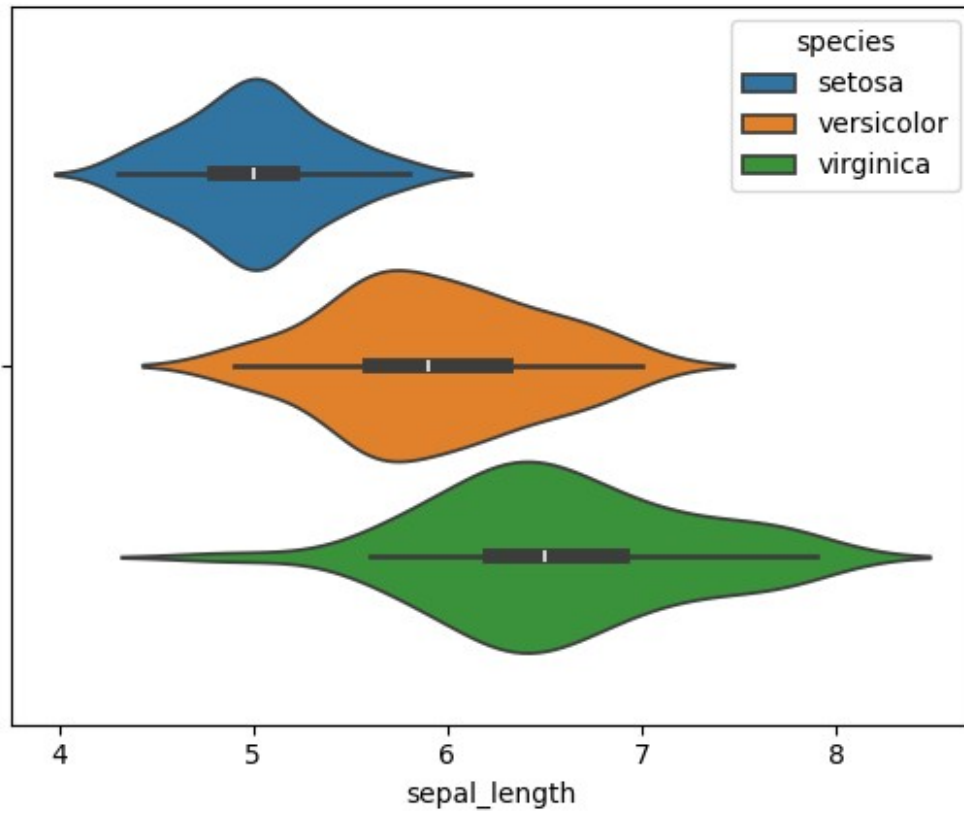


```
phool.columns
```

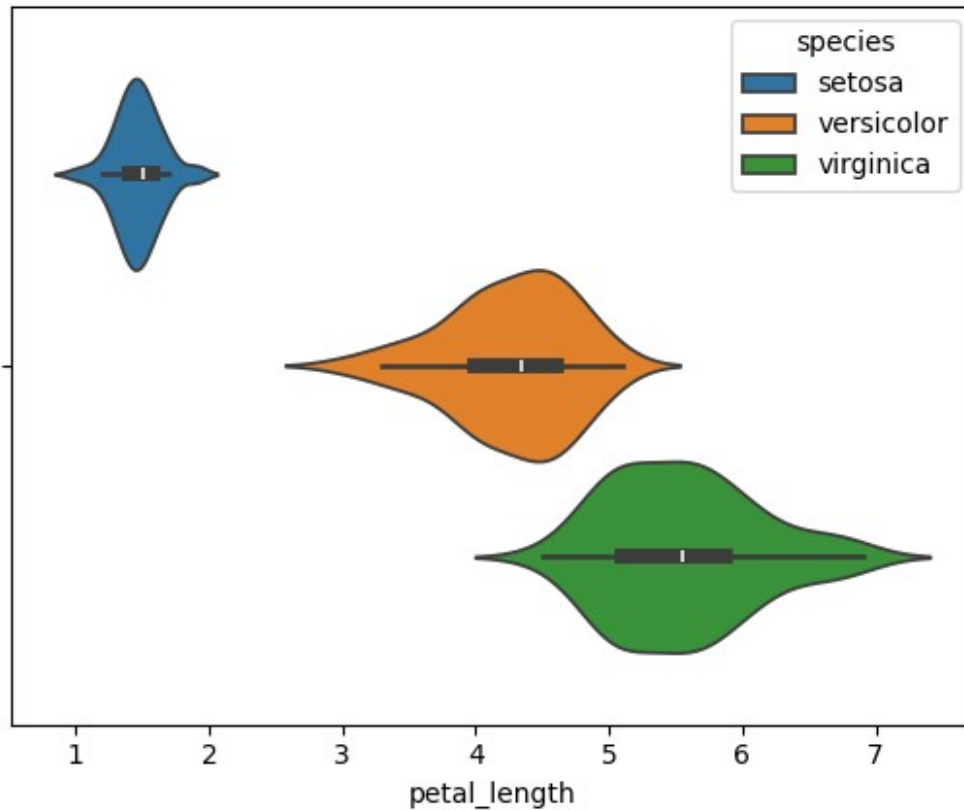
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',  
      'species',  
      'sepal_width_category'],  
      dtype='object')
```

```
sns.violinplot(data=phool, x="sepal_length", hue="species")
```

```
<Axes: xlabel='sepal_length'>
```

```
sns.violinplot(data=phool, x="petal_length", hue="species")  
<Axes: xlabel='petal_length'>
```



```
# error in this on iris dataset

# # heat map
# corr = phool.corr()
# plt.figure(figsize=(10,10))
# sns.heatmap(corr)
# plt.title("Heat ma of kashti Data")

# this method work

numeric_columns = phool.select_dtypes(include=['float64',
'int64']).columns
corr = phool[numeric_columns].corr()

# Set the size of the heatmap figure
plt.figure(figsize=(10, 10))

# Create a heatmap using Seaborn
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)

# Set the title
```

```
plt.title("Heatmap of phool Data - Correlation Matrix")
```

```
Text(0.5, 1.0, 'Heatmap of phool Data - Correlation Matrix')
```

