

iris_plot_day11_assignment1

January 11, 2023

1 Plotting

1- Import required libraries

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

2- Import datasets

```
[ ]: iris = sns.load_dataset('iris')
iris
```

```
[ ]:      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]

```
[ ]: iris.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
```

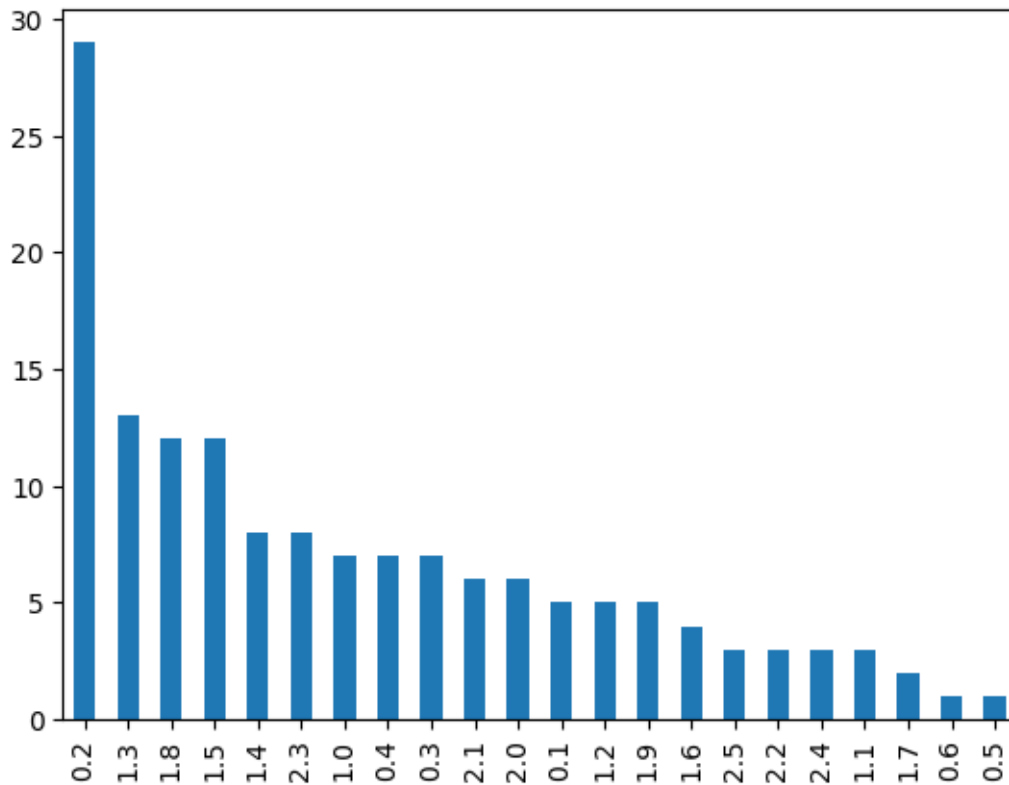
```
3   petal_width    150 non-null    float64
4   species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
[ ]: iris.isnull().sum()
```

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

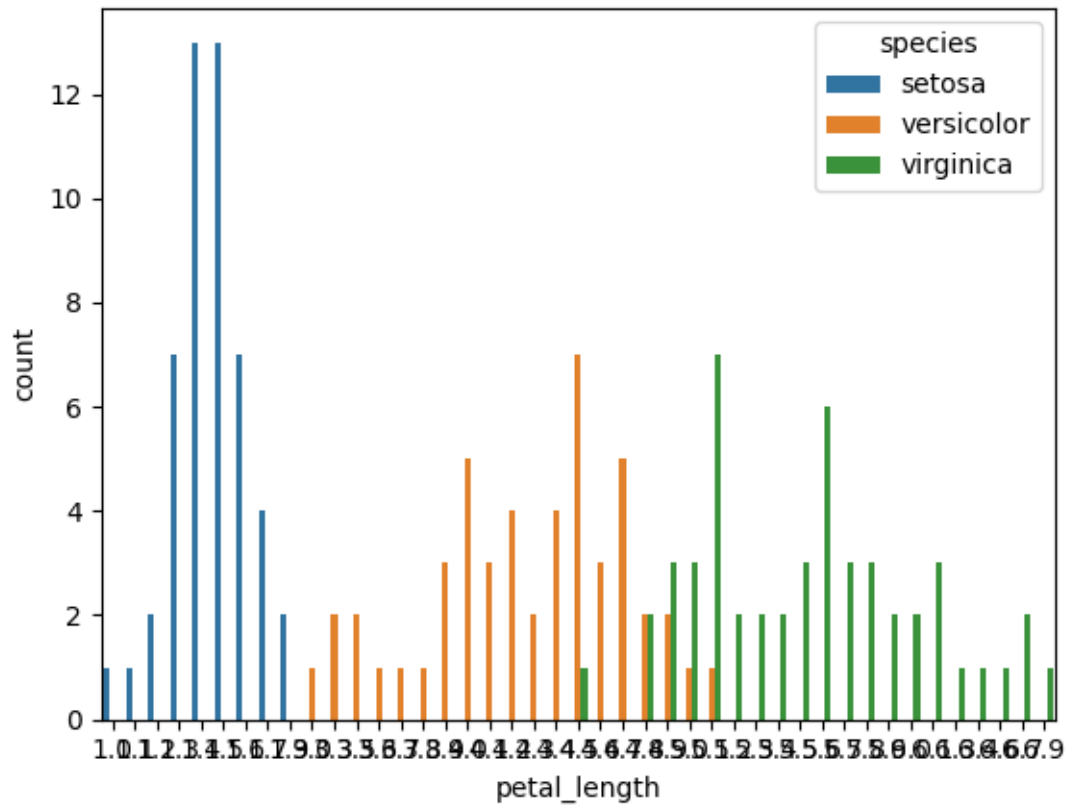
```
[ ]: iris.petal_width.value_counts().plot(kind='bar')
```

```
[ ]: <AxesSubplot: >
```



```
[ ]: sns.countplot(x='petal_length', hue = 'species', data = iris)
```

```
[ ]: <AxesSubplot: xlabel='petal_length', ylabel='count'>
```



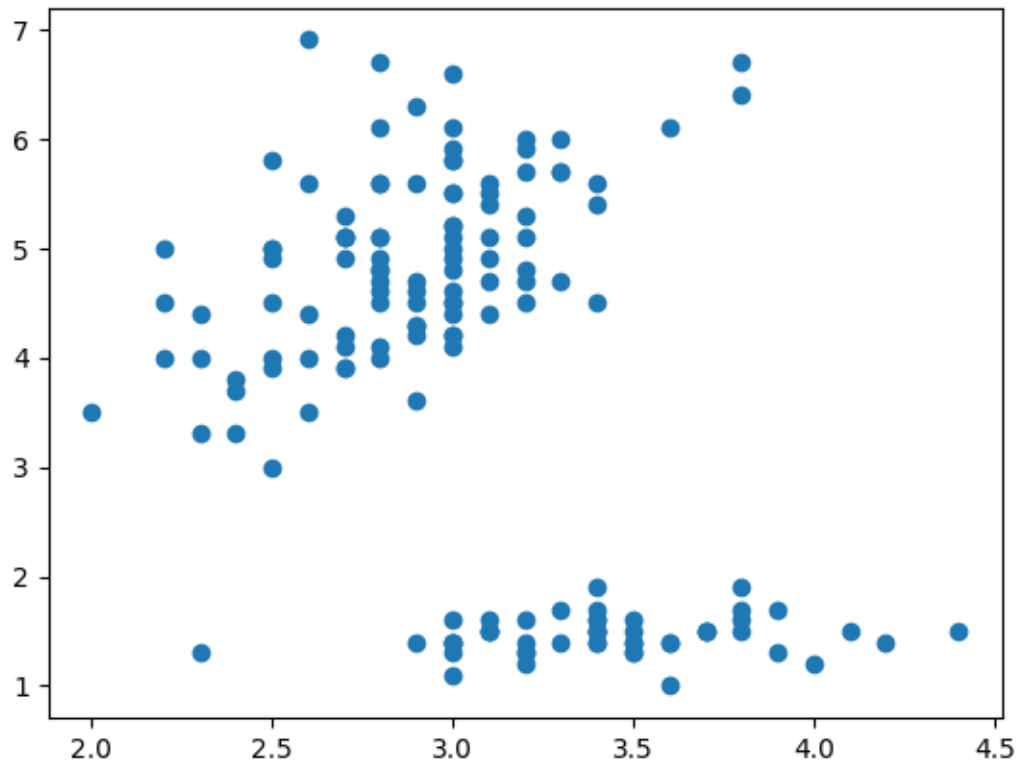
```
[ ]: iris.columns
```

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

```
[ ]: #scatter Plot
```

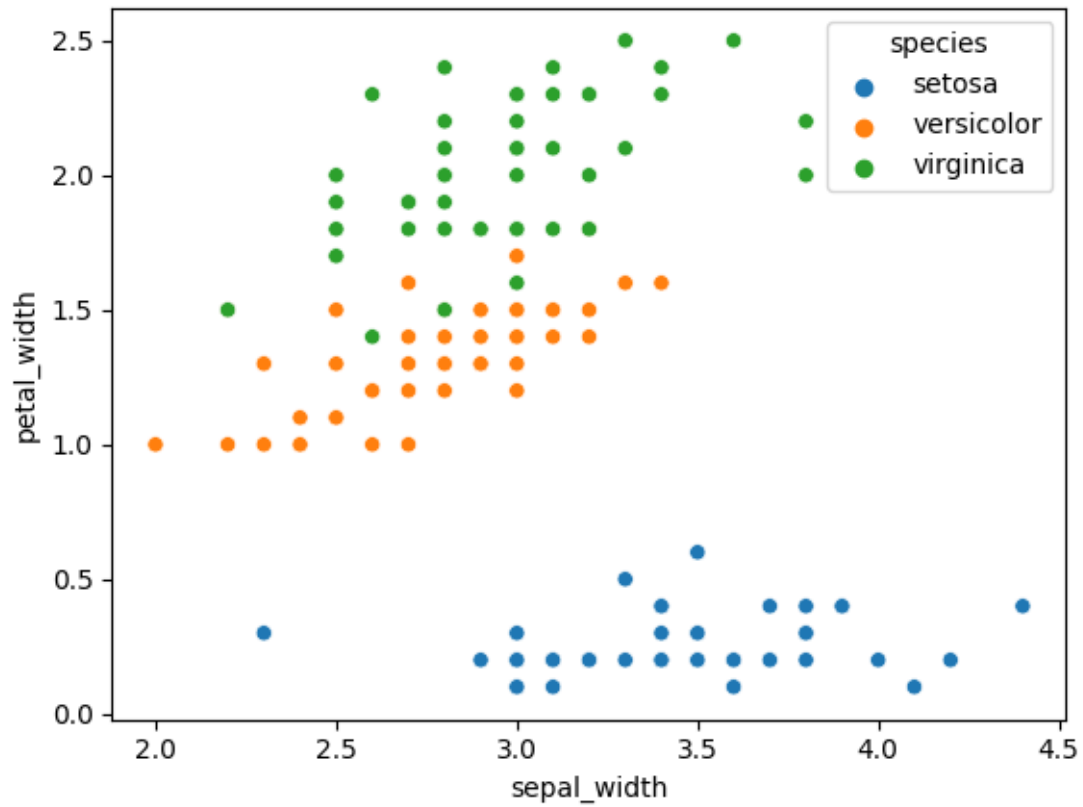
```
plt.scatter(data=iris, x='sepal_width', y = 'petal_length')
```

```
[ ]: <matplotlib.collections.PathCollection at 0x27ba01e3850>
```



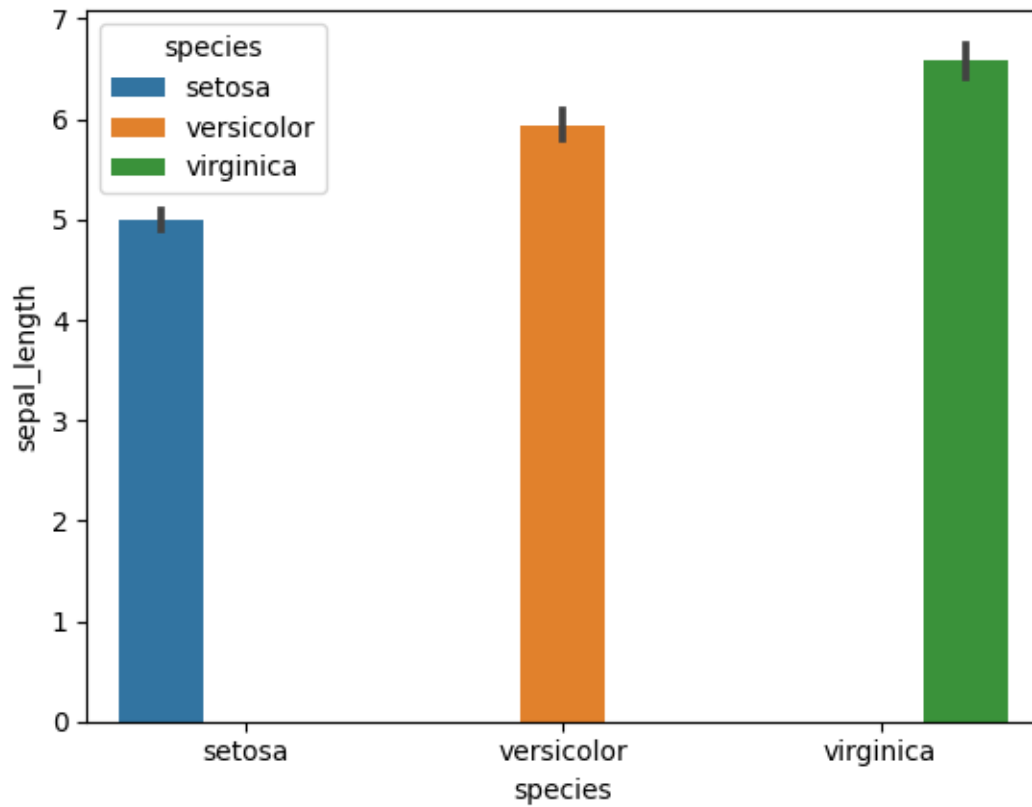
```
[ ]: sns.scatterplot(data=iris, x='sepal_width', y = 'petal_width', hue = 'species')
```

```
[ ]: <AxesSubplot: xlabel='sepal_width', ylabel='petal_width'>
```



```
[ ]: sns.barplot(data=iris, x='species', y = 'sepal_length', hue = 'species')
```

```
[ ]: <AxesSubplot: xlabel='species', ylabel='sepal_length'>
```

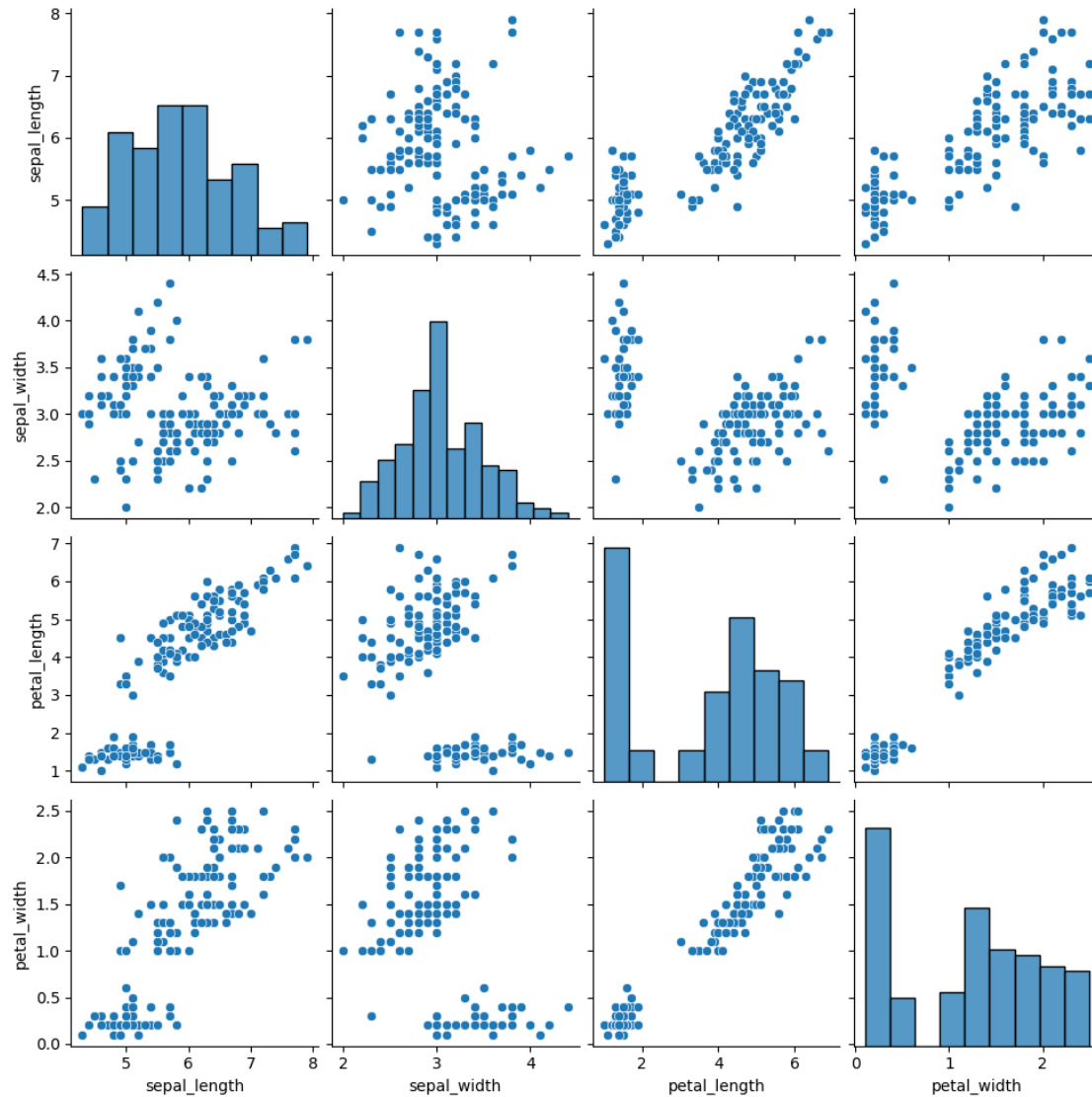


```
[ ]: iris.columns
```

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

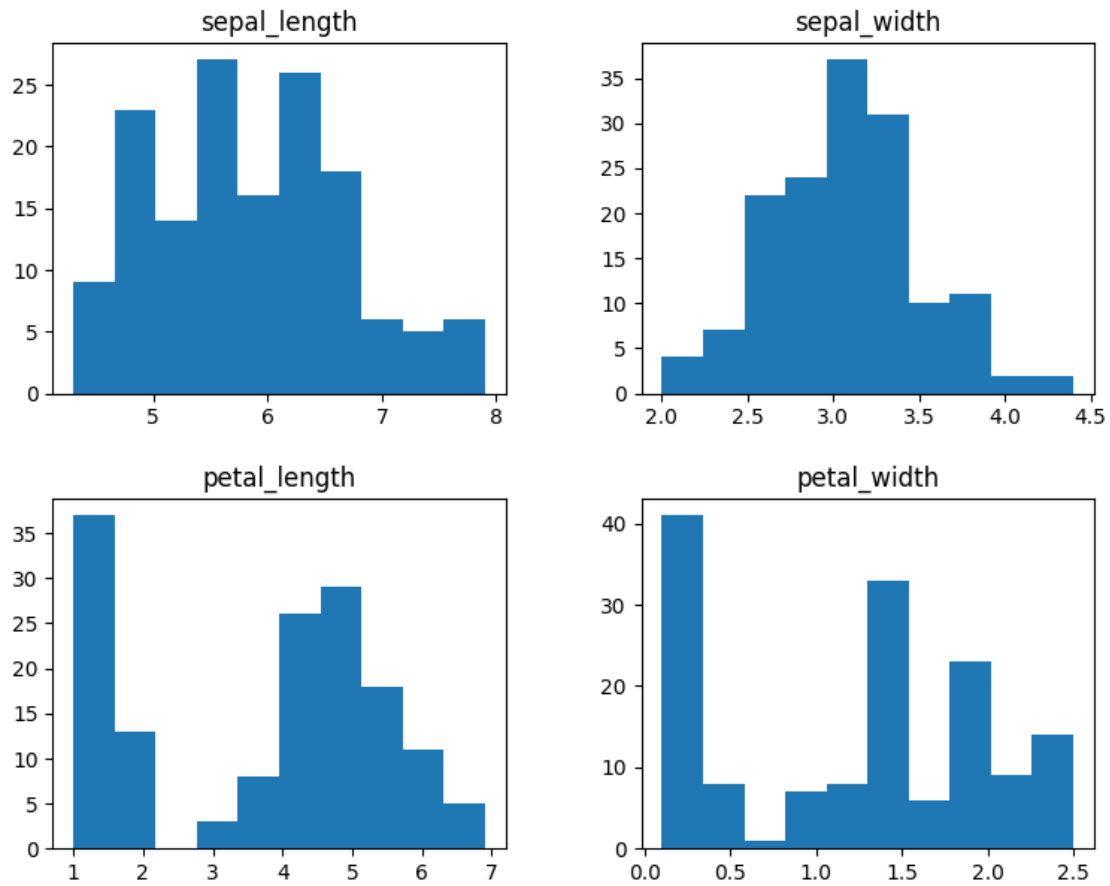
```
[ ]: sns.pairplot(iris)
```

```
[ ]: <seaborn.axisgrid.PairGrid at 0x27ba06d6550>
```



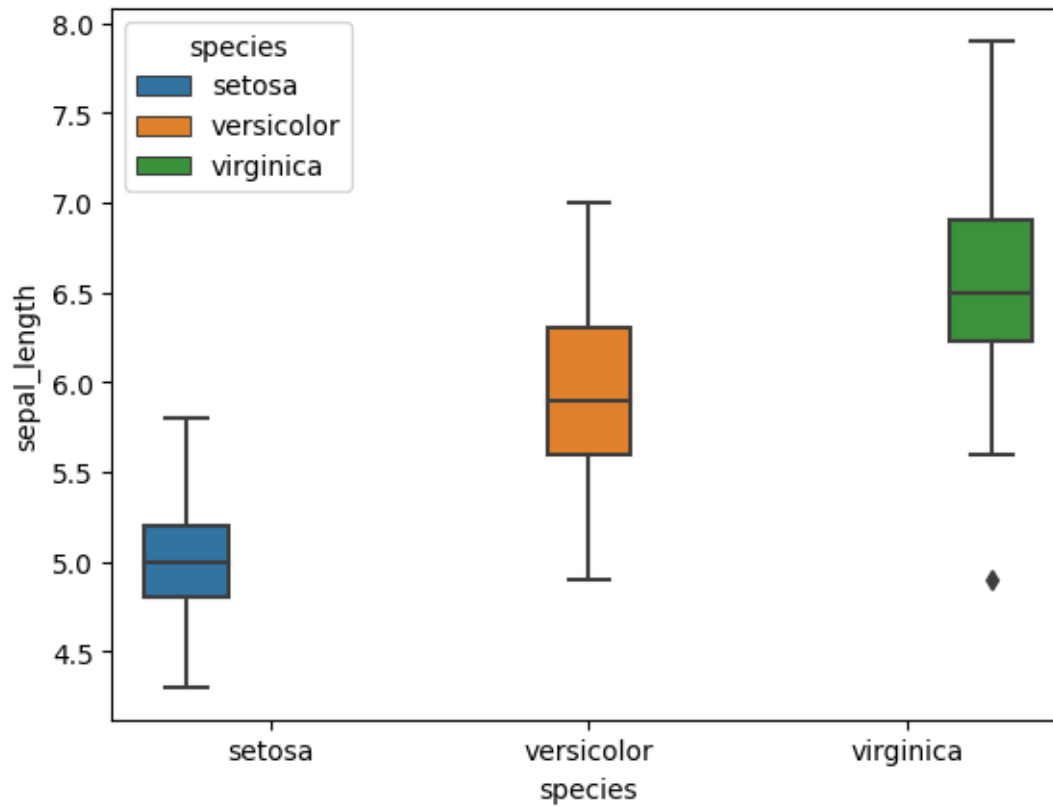
```
[ ]: iris.hist(bins = 10, figsize=(9,7), grid=False)
```

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



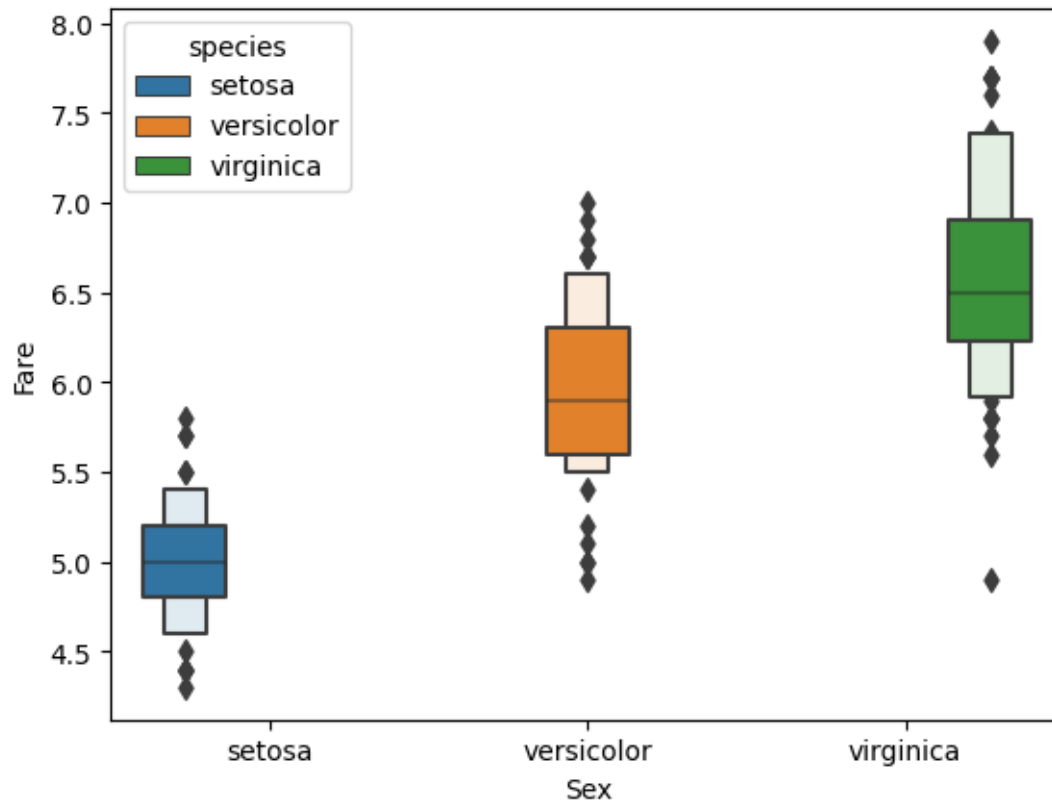
```
[ ]: sns.boxplot(data=iris, x='species', y = 'sepal_length', hue = 'species')
```

```
[ ]: <AxesSubplot: xlabel='species', ylabel='sepal_length'>
```

```
[ ]: #how to change x and y axes title
sns.boxenplot(data=iris, x='species', y = 'sepal_length', hue = 'species')
plt.xlabel("Sex")
plt.ylabel("Fare")
```

```
[ ]: Text(0, 0.5, 'Fare')
```



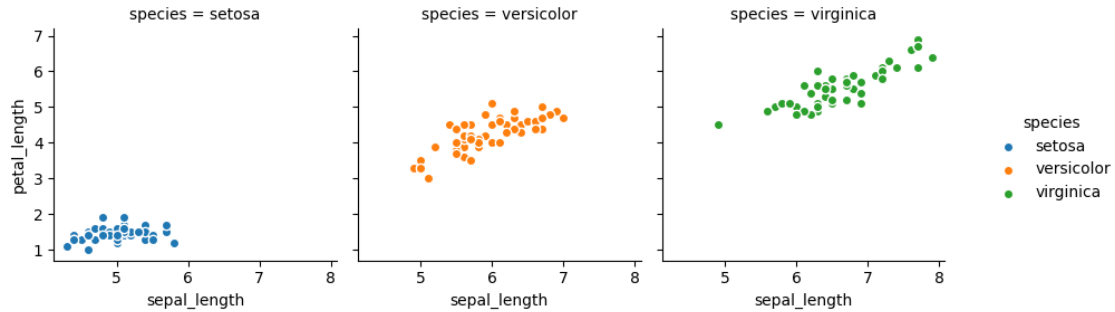
```
[ ]: iris
```

```
[ ]:      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]
```

```
[ ]: g= sns.FacetGrid(iris, hue='species', col='species', margin_titles=True)
#      palette=(1:'seagreen', 0:'gray'))
```

```
g= g.map(plt.scatter, 'sepal_length', 'petal_length', edgecolor= 'w').
    ↪add_legend();
```



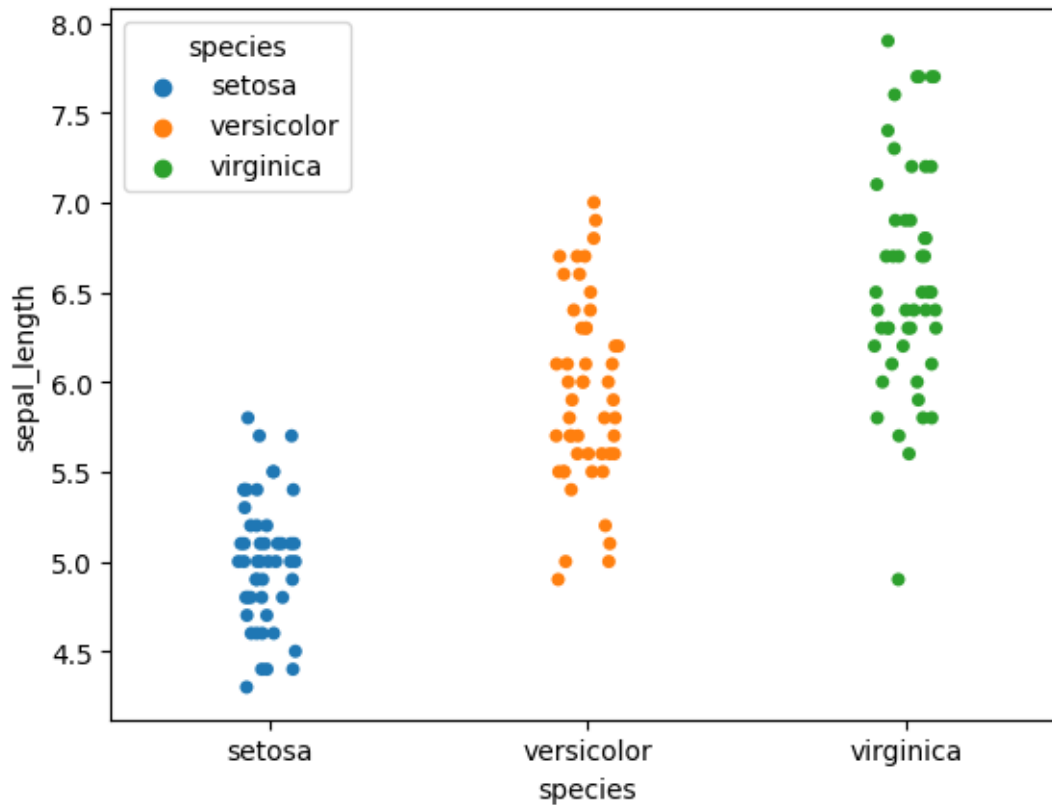
```
[ ]: s=sns.FacetGrid(tit, hue='survived', col='pclass', margin_titles=True,
    palette = (1:'seagreen', 0:'gray'))
```

```
Cell In[45], line 2
    palette = (1:'seagreen', 0:'gray'))
```

SyntaxError: invalid syntax

```
[ ]: sns.stripplot(data=iris, x='species', y = 'sepal_length', hue = 'species')
```

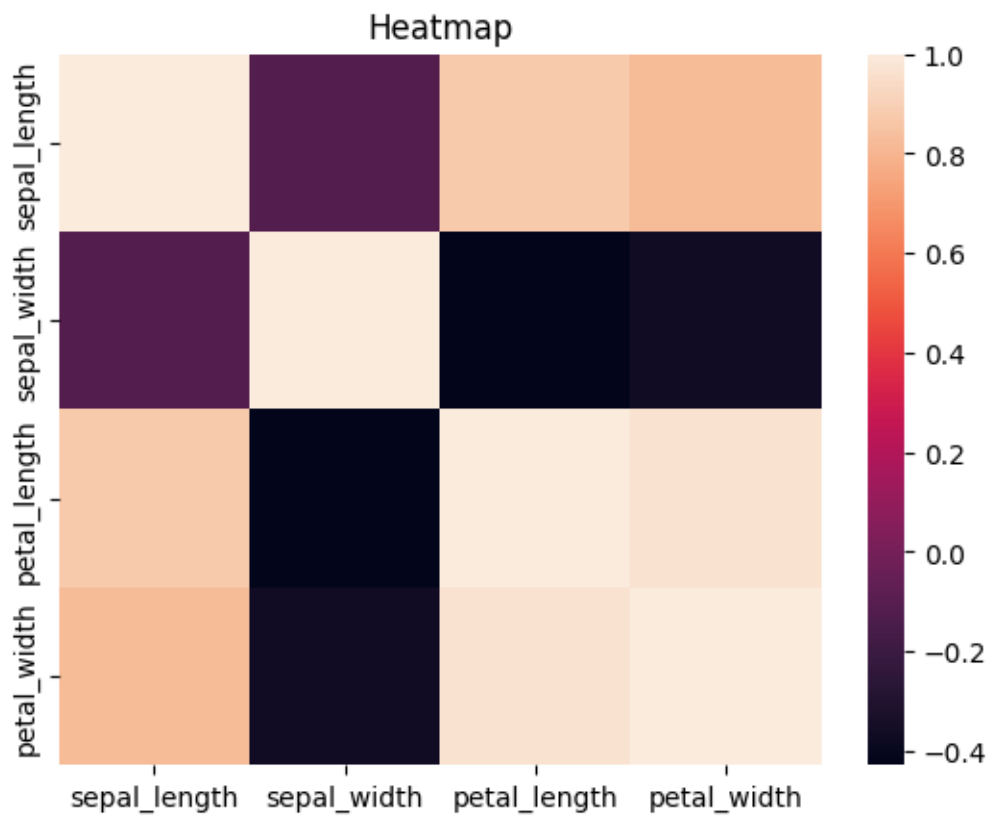
```
[ ]: <AxesSubplot: xlabel='species', ylabel='sepal_length'>
```



```
[ ]: #Heatmap
corr = iris.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr)
plt.title('Heatmap')
```

C:\Users\muham\AppData\Local\Temp\ipykernel_19036\532938971.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
corr = iris.corr()

```
[ ]: Text(0.5, 1.0, 'Heatmap')
```



```
[ ]: sns.violinplot(data=iris, x='species', y = 'sepal_length', hue = 'species')
```

```
[ ]: <AxesSubplot: xlabel='species', ylabel='sepal_length'>
```

