Programs using matplotlib / seaborn for data visualisation

a)Write a program to draw univariate visualization plots(line plot,histogram,boxplot,barchart,piechart) with matplotlib for iris dataset

In [54]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:
```
i = pd.read_csv('iris.csv')
df = pd.DataFrame(i)
print(df)
```

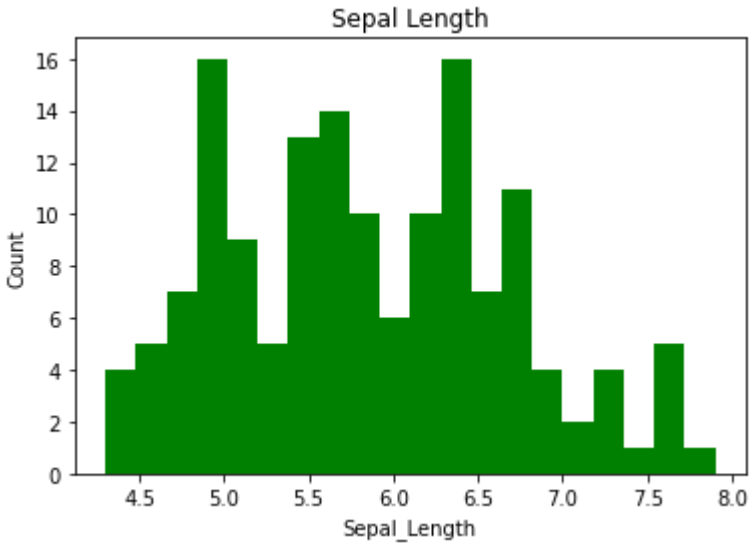
```
     sepal_length  sepal_width  petal_length  petal_width    specie
s
0             5.1          3.5           1.4          0.2     setos
a
1             4.9          3.0           1.4          0.2     setos
a
2             4.7          3.2           1.3          0.2     setos
a
3             4.6          3.1           1.5          0.2     setos
a
4             5.0          3.6           1.4          0.2     setos
a
..            ...          ...           ...          ...
...
145           6.7          3.0           5.2          2.3  virginic
a
146           6.3          2.5           5.0          1.9  virginic
a
147           6.5          3.0           5.2          2.0  virginic
a
148           6.2          3.4           5.4          2.3  virginic
a
149           5.9          3.0           5.1          1.8  virginic
a

[150 rows x 5 columns]
```

Histogram

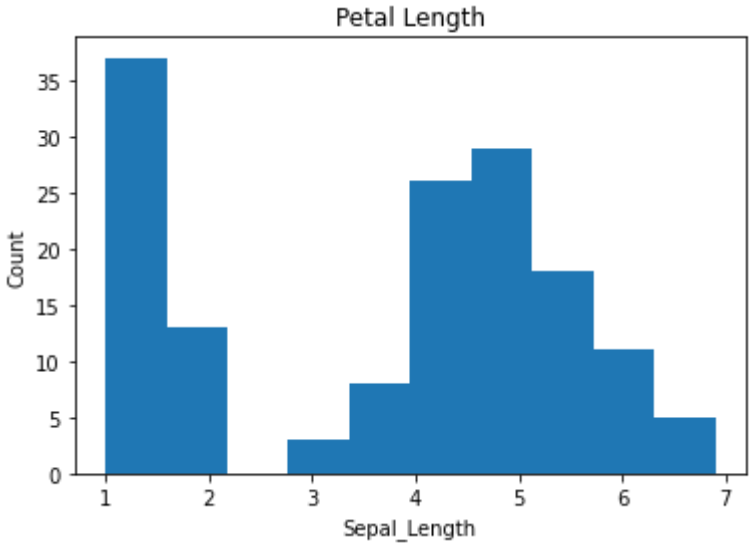
In [76]:
```
x = df["sepal_length"]
plt.hist(x, bins = 20, color = "g")
plt.title("Sepal Length")
plt.xlabel("Sepal_Length")
plt.ylabel("Count")
```

Out[76]:
```
Text(0, 0.5, 'Count')
```



In [69]:
```
df["petal_length"].plot(kind = 'hist')
plt.title("Petal Length")
plt.xlabel("Sepal_Length")
plt.ylabel("Count")
```

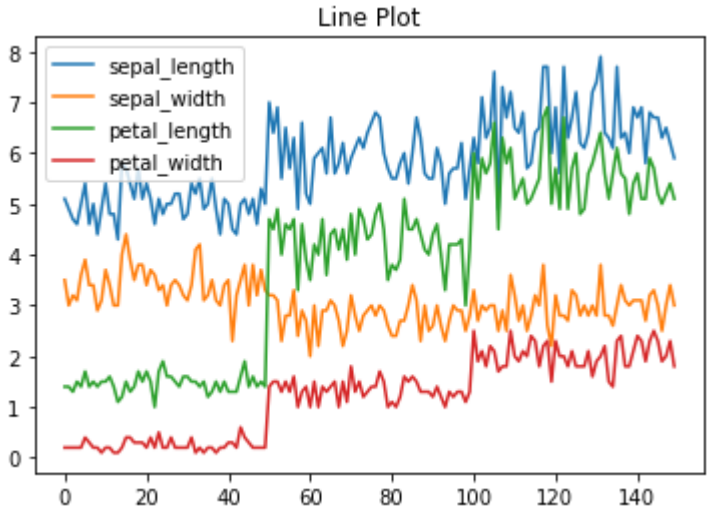
Out[69]:
```
Text(0, 0.5, 'Count')
```



Line

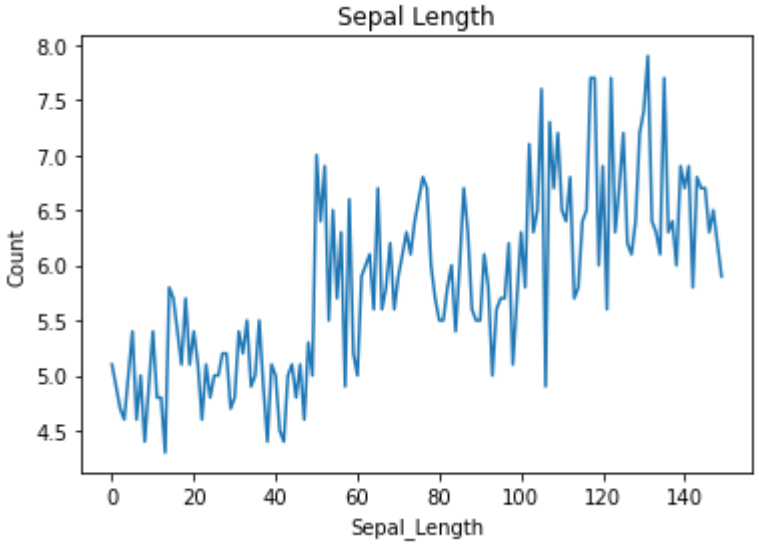
In [70]:
```
df.plot(kind = 'line')
plt.title("Line Plot")
```

Out[70]:
```
Text(0.5, 1.0, 'Line Plot')
```



In [21]:
```
df["sepal_length"].plot(kind = 'line')
plt.title("Sepal Length")
plt.xlabel("Sepal_Length")
plt.ylabel("Count")
```

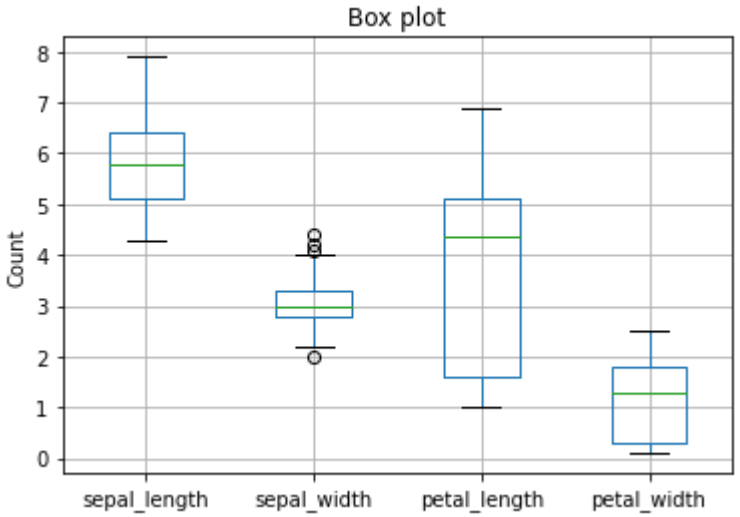
Out[21]:
```
Text(0, 0.5, 'Count')
```



Box

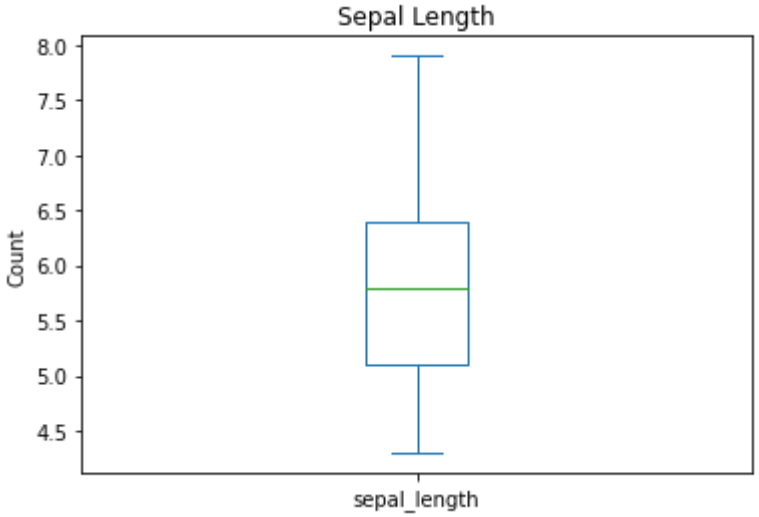
In [73]:
```
df.boxplot()
plt.ylabel("Count")
plt.title("Box plot")
```

Out[73]:
```
Text(0.5, 1.0, 'Box plot')
```



In [71]:
```
df["sepal_length"].plot(kind = 'box')
plt.title("Sepal Length")
#plt.xlabel("Sepal_Length")
plt.ylabel("Count")
```

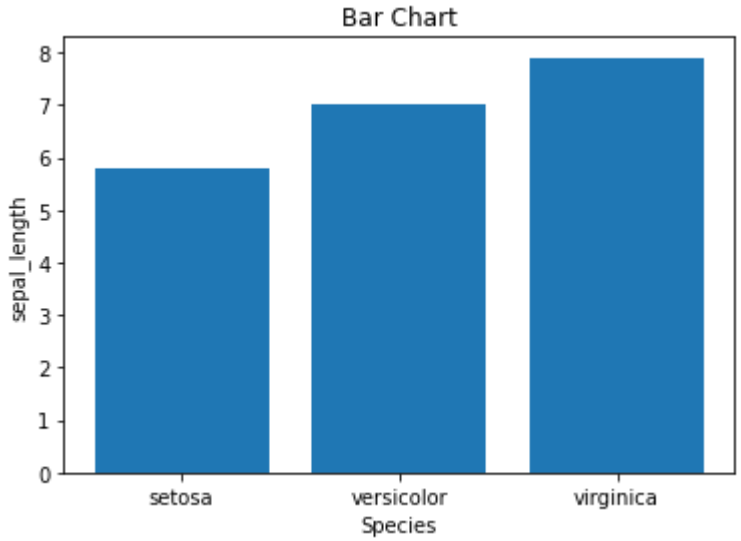
Out[71]:
```
Text(0, 0.5, 'Count')
```



Barplot

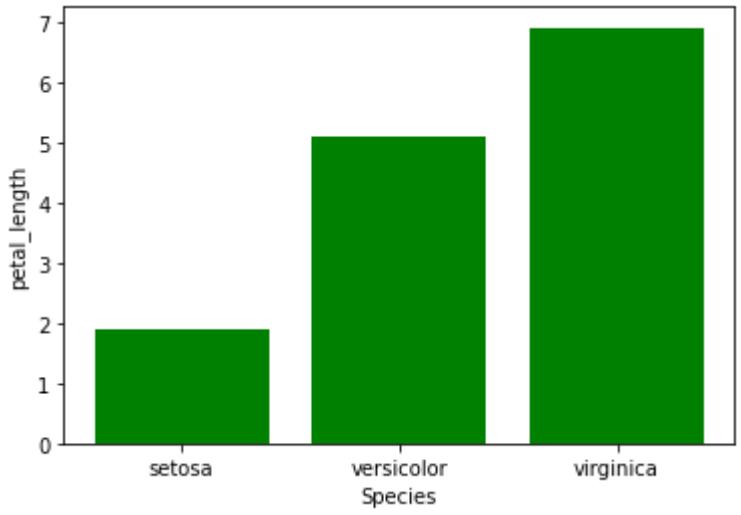
In [77]:
```
plt.bar(df['species'],df['sepal_length'])
plt.title('Bar Chart')
plt.xlabel('Species')
plt.ylabel('sepal_length')
```

Out[77]:
```
Text(0, 0.5, 'sepal_length')
```



In [84]:
```
plt.bar(df['species'],df['petal_length'], color = 'g')
plt.xlabel('Species')
plt.ylabel('petal_length')
```

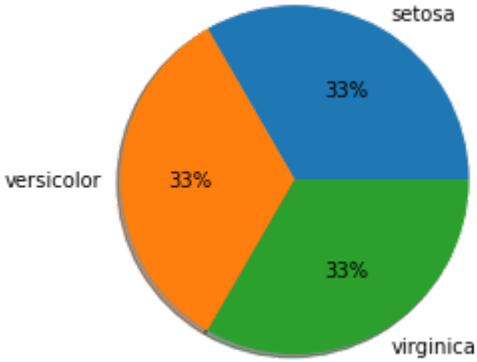
Out[84]:
```
Text(0, 0.5, 'petal_length')
```



Pie

In [88]:
```
plt.pie(df['species'].value_counts(),labels=['setosa','versicolor','
```

Out[88]:
```
([<matplotlib.patches.Wedge at 0x7fe312ad0e20>,
  <matplotlib.patches.Wedge at 0x7fe312adc7f0>,
  <matplotlib.patches.Wedge at 0x7fe312ae81c0>],
 [Text(0.5499999702695115, 0.9526279613277875, 'setosa'),
  Text(-1.0999999999999954, -1.0298943258065002e-07, 'versicolor'),
  Text(0.5500001486524352, -0.9526278583383436, 'virginica')],
 [Text(0.2999999837833699, 0.5196152516333385, '33%'),
  Text(-0.5999999999999974, -5.6176054134900006e-08, '33%'),
  Text(0.30000008108314646, -0.5196151954572783, '33%')])
```

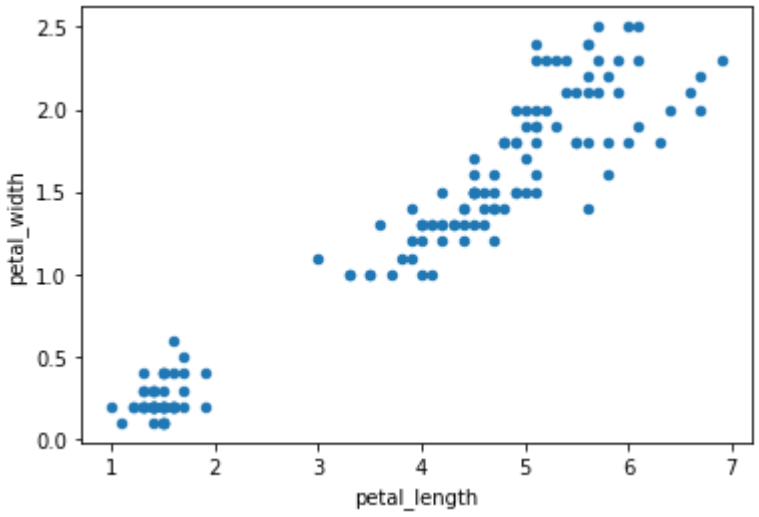


b)Write a program to draw multivariate visualization plots(scatter plots, scatter multiple,scatter matrix,bubble plot) with matplotlib for iris dataset

Scatter plot

In [42]:
```
df.plot.scatter(x='petal_length', y = 'petal_width')
```

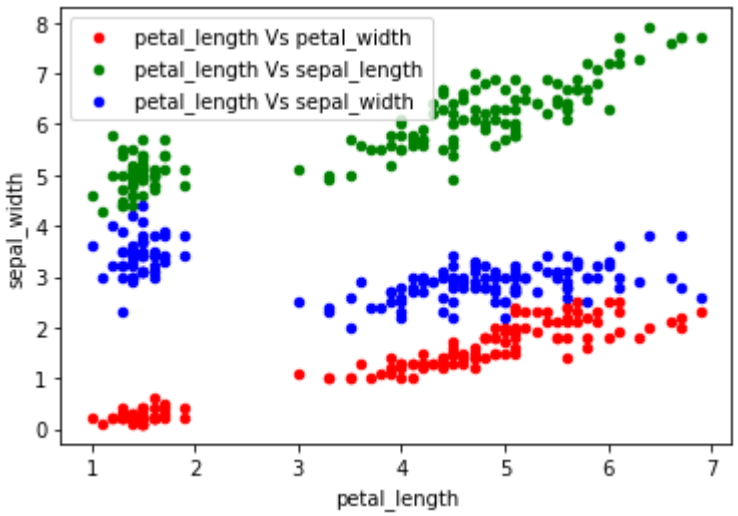
Out[42]:
```
<AxesSubplot:xlabel='petal_length', ylabel='petal_width'>
```



Scatter Multiple

In [44]:
```
ax1 = df.plot(kind='scatter', x='petal_length', y = 'petal_width', la
ax2 = df.plot(kind='scatter', x='petal_length', y = 'sepal_length', l
ax3 = df.plot(kind='scatter',  x='petal_length', y = 'sepal_width',la
print(ax1 == ax2 == ax3)
```

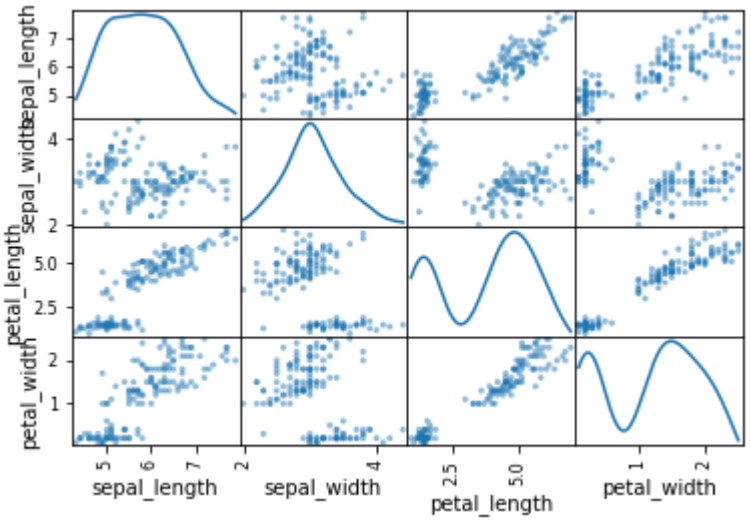
```
True
```

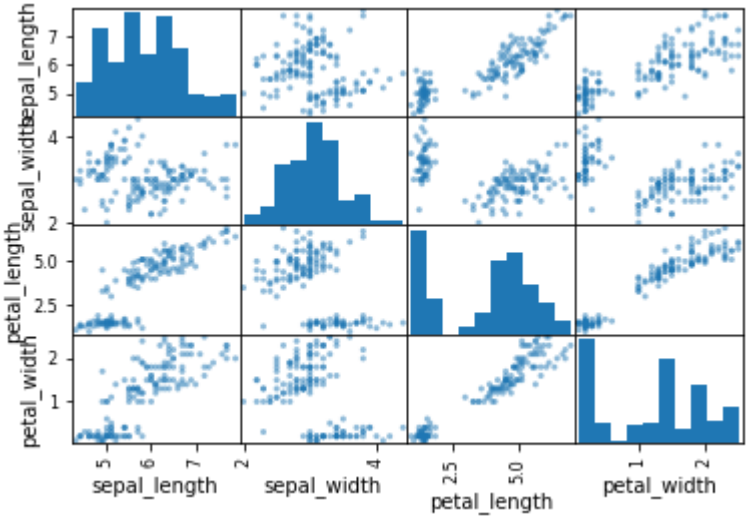


Scatter Matrix

In [47]:
```
from pandas.plotting import scatter_matrix
scatter_matrix(df, alpha=0.5, diagonal='kde')
pd.plotting.scatter_matrix(df)
```

Out[47]:
```
array([[<AxesSubplot:xlabel='sepal_length', ylabel='sepal_length'>,
        <AxesSubplot:xlabel='sepal_width', ylabel='sepal_length'>,
        <AxesSubplot:xlabel='petal_length', ylabel='sepal_length'>,
        <AxesSubplot:xlabel='petal_width', ylabel='sepal_length'>],
       [<AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>,
        <AxesSubplot:xlabel='sepal_width', ylabel='sepal_width'>,
        <AxesSubplot:xlabel='petal_length', ylabel='sepal_width'>,
        <AxesSubplot:xlabel='petal_width', ylabel='sepal_width'>],
       [<AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>,
        <AxesSubplot:xlabel='sepal_width', ylabel='petal_length'>,
        <AxesSubplot:xlabel='petal_length', ylabel='petal_length'>,
        <AxesSubplot:xlabel='petal_width', ylabel='petal_length'>],
       [<AxesSubplot:xlabel='sepal_length', ylabel='petal_width'>,
        <AxesSubplot:xlabel='sepal_width', ylabel='petal_width'>,
        <AxesSubplot:xlabel='petal_length', ylabel='petal_width'>,
        <AxesSubplot:xlabel='petal_width', ylabel='petal_width'>]],
      dtype=object)
```

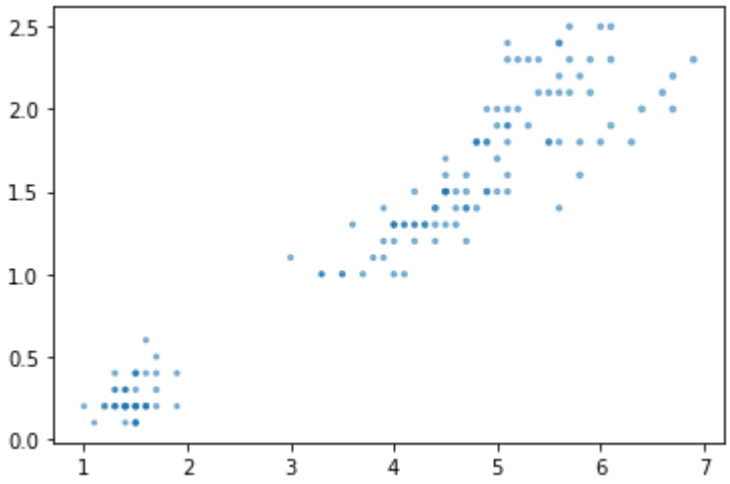




Bubble Plot

In [53]:
```
plt.scatter('petal_length', 'petal_width',  s='sepal_length',  alpha=
```

Out[53]:
```
<matplotlib.collections.PathCollection at 0x7fe3150f9d30>
```

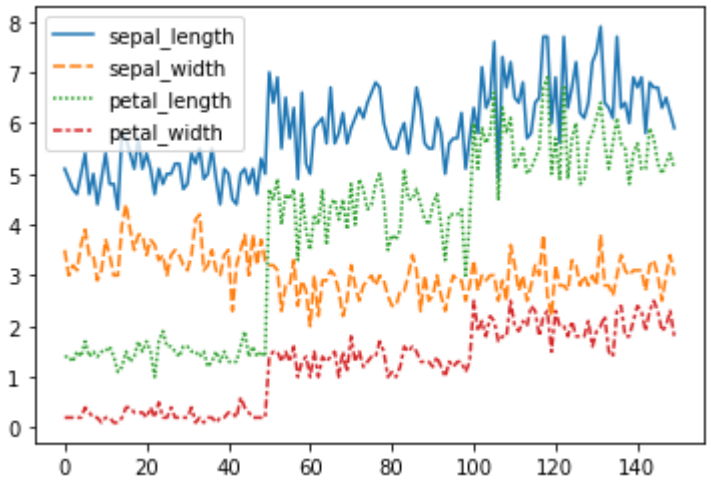


c)Write a program to draw univariate and multivariate visualization plots with seaborn(line plot, pairplot,jointplot,heatmap) for iris dataset

Line plot

In [57]:
```
sns.lineplot(data=df)
```

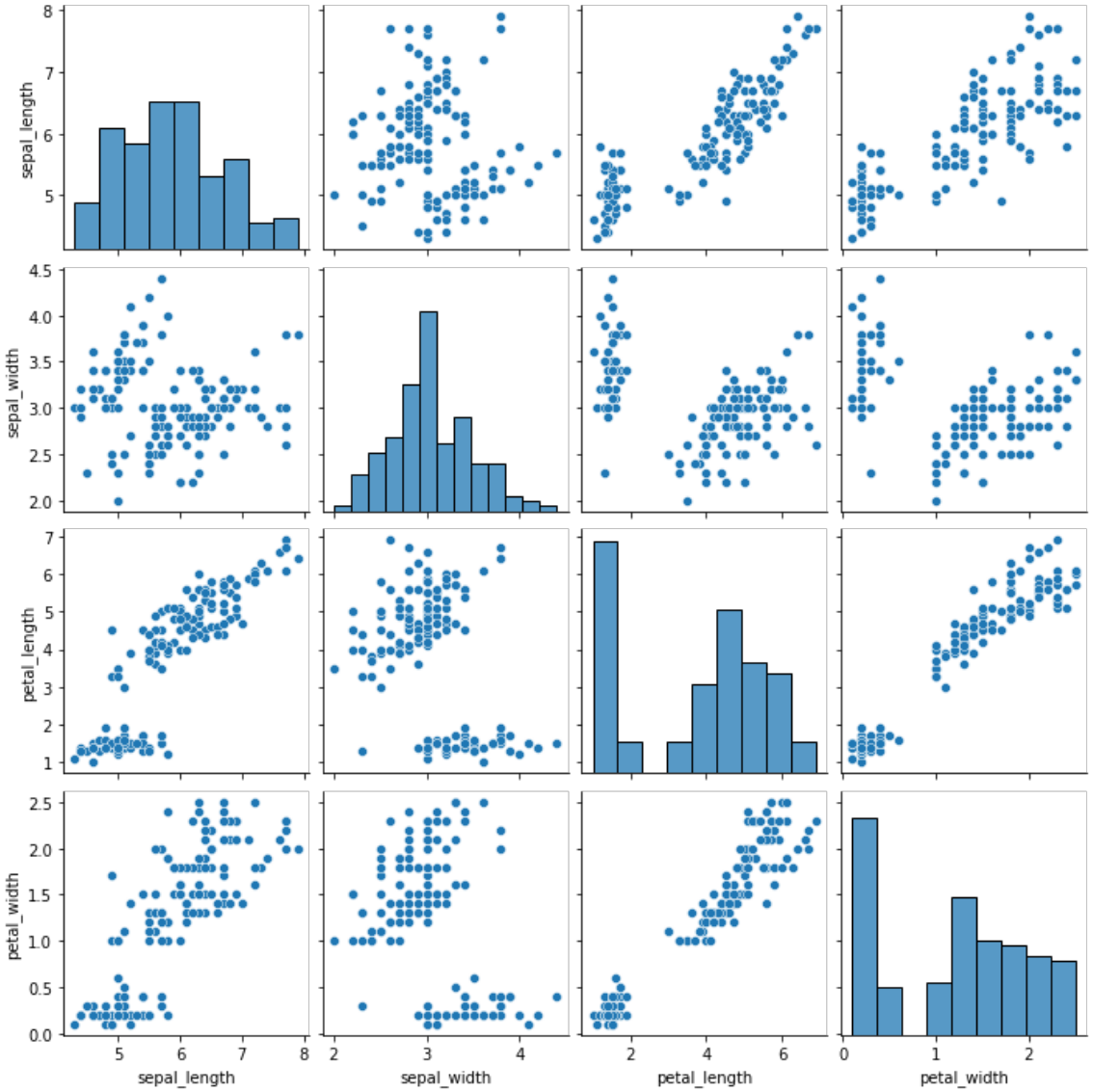
Out[57]:
```
<AxesSubplot:>
```



Pair Plot

In [59]:
```
sns.pairplot(data=df,kind='scatter')
```

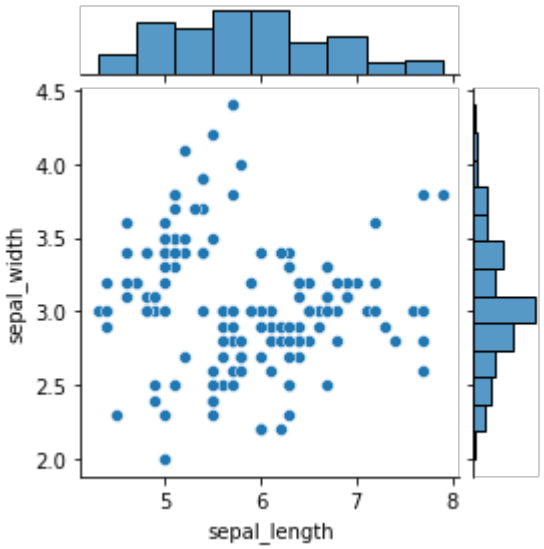
Out[59]:
```
<seaborn.axisgrid.PairGrid at 0x7fe314b0e490>
```



Joint Plot

In [58]:
```
sns.jointplot(x='sepal_length',y='sepal_width',data=df,height=4)
```

Out[58]:
```
<seaborn.axisgrid.JointGrid at 0x7fe314c604c0>
```



Heat Map

In [63]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[63]:
```
<AxesSubplot:>
```

Boxplot

In [91]:
```
p = sns.boxplot(data=iris)
p.set_ylabel("count")
```

Out[91]:
```
Text(0, 0.5, 'count')
```