Program 23

October 25, 2022

1 PROGRAM 23

Aim : 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 b)Write a program to draw multivariate visualization plots(scatter plots, scatter multiple,scatter matrix,bubble plot) with matplotlib for iris dataset c)Write a program to draw univariate and multivariate visualization plots with seaborn(line plot, pairplot,jointplot,heatmap) for iris dataset ##### Date : 14/09/2022 ##### By : Anu C Scharia

[5]:
```
import matplotlib.pyplot as plt
```

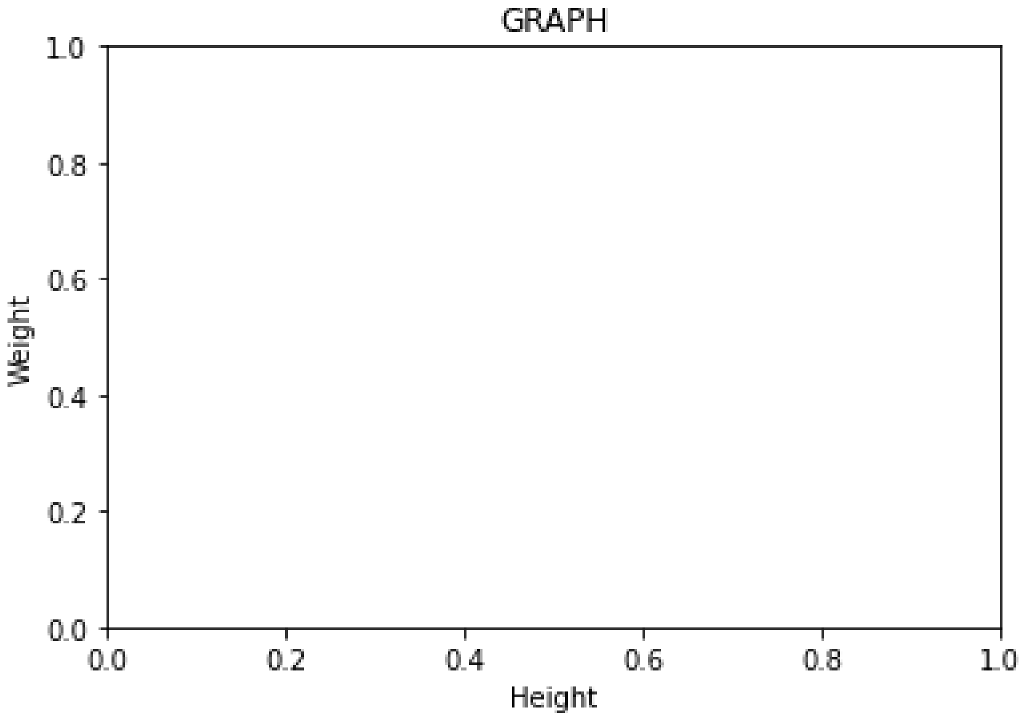
[10]:
```
plt.figure(figsize = (10, 7))
```

[10]:
```
<Figure size 720x504 with 0 Axes>
```

```
<Figure size 720x504 with 0 Axes>
```

[30]:
```
plt.title("GRAPH")
plt.xlabel("Height")
plt.ylabel("Weight")
```

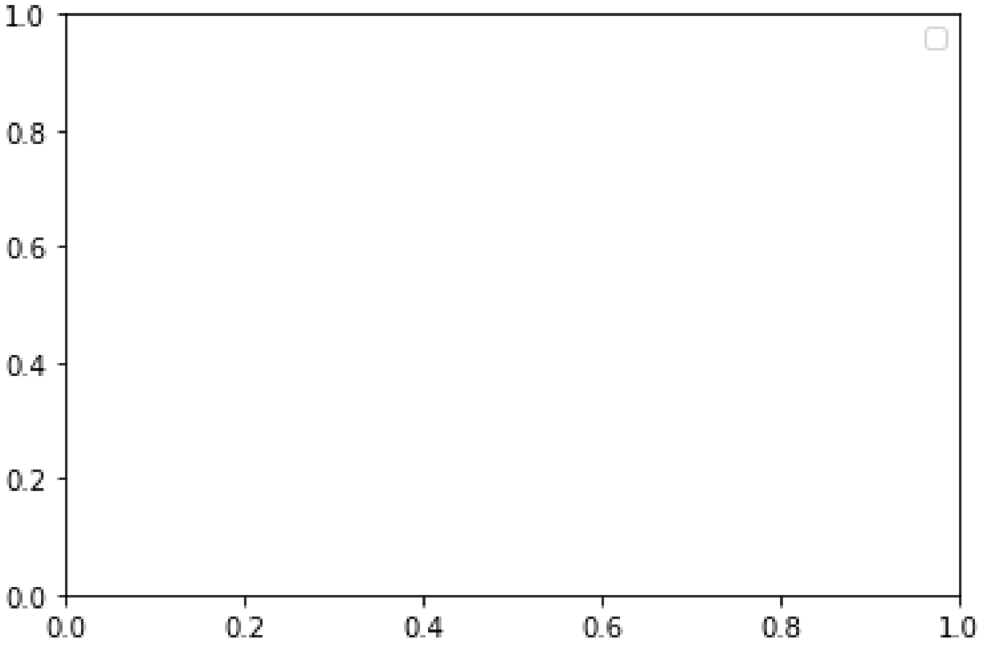
[30]:
```
Text(0, 0.5, 'Weight')
```



```
[15]: plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```

```
[15]: <matplotlib.legend.Legend at 0x7f75f0dac6d0>
```



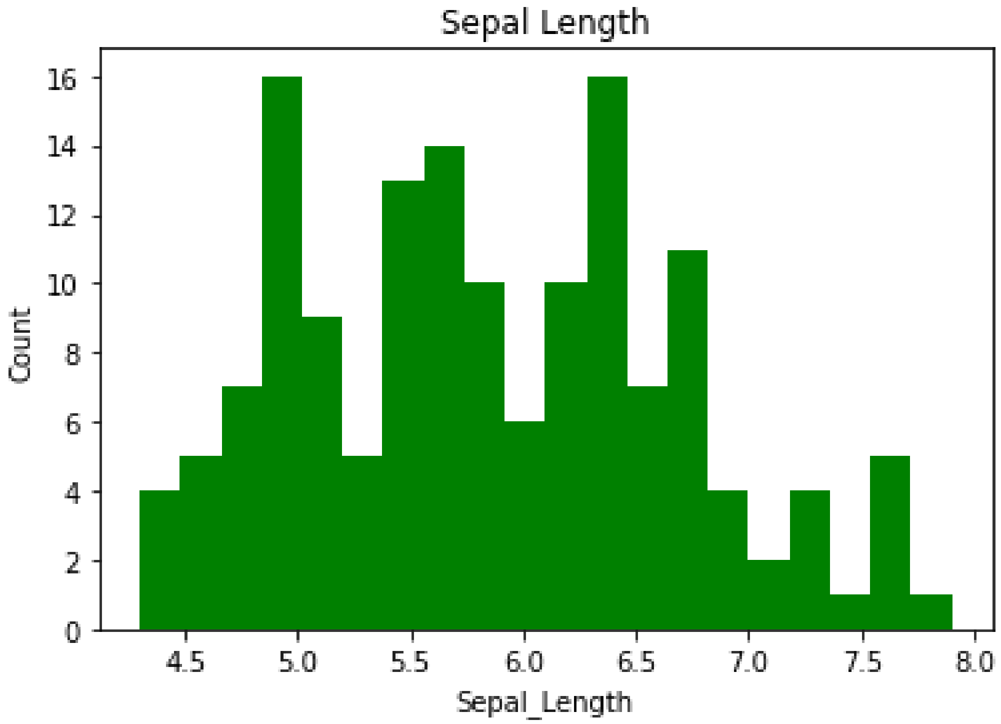
[26]:
```
plt.show()
```

[3]:
```
import pandas as pd
df=pd.read_csv('iris.csv')
df=pd.DataFrame(df)
x=df["sepal_length"]
```

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

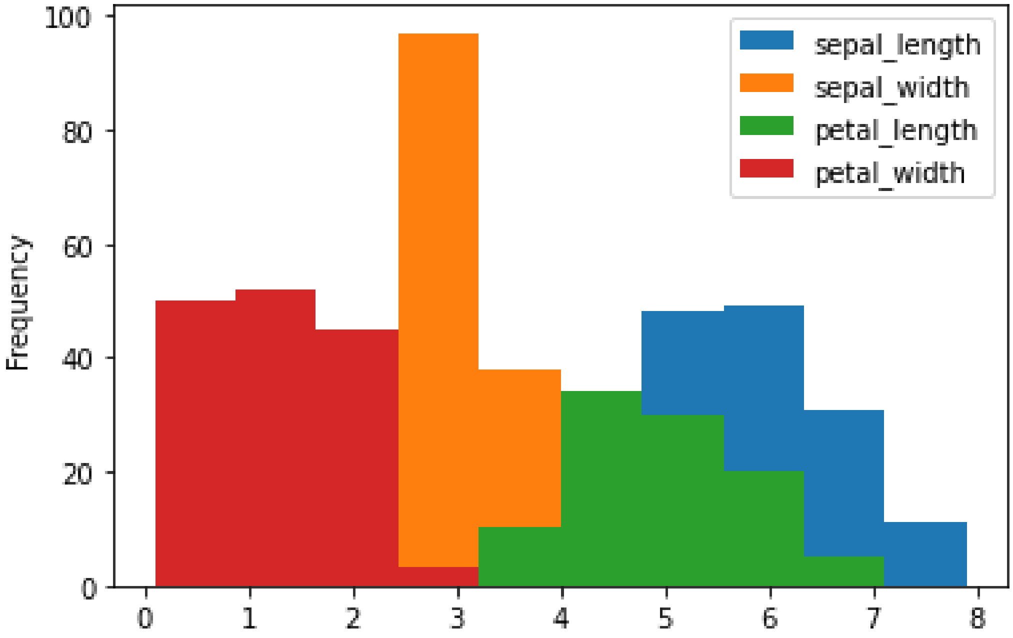
[86]:
```
plt.hist(x, bins = 20, color = "green")
plt.title("Sepal Length")
plt.xlabel("Sepal_Length")
plt.ylabel("Count")
```

[86]:
```
Text(0, 0.5, 'Count')
```



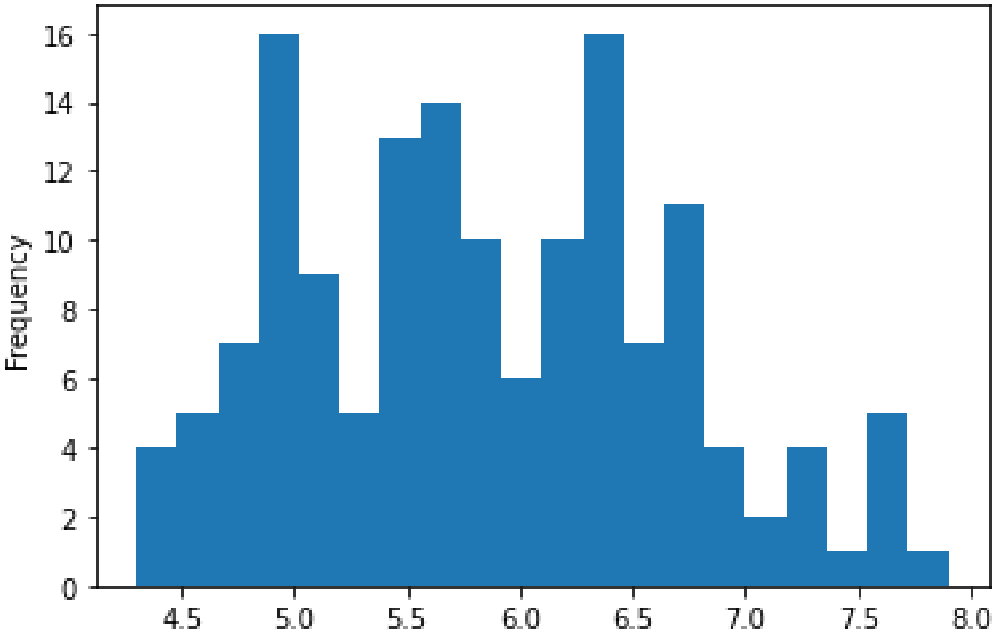
```
[19]: #histogram
      df.plot(kind='hist')
```

```
[19]: <AxesSubplot:ylabel='Frequency'>
```



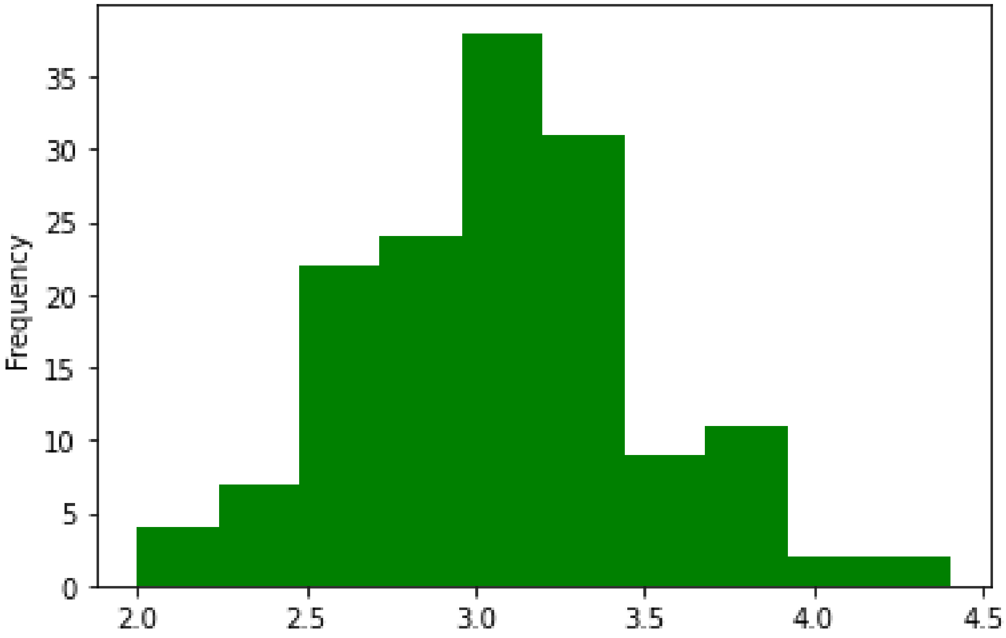
[31]:
```
df["sepal_length"].plot(kind='hist',bins = 20)
```

[31]:
```
<AxesSubplot:ylabel='Frequency'>
```



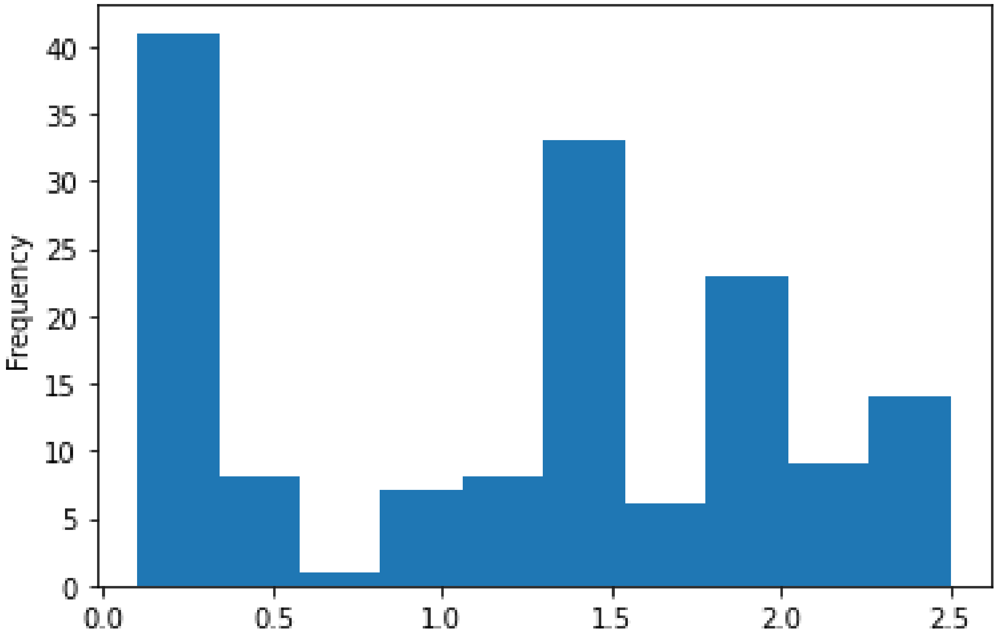
[32]:
```
df["sepal_width"].plot(kind='hist', color = "green")
```

[32]:
```
<AxesSubplot:ylabel='Frequency'>
```



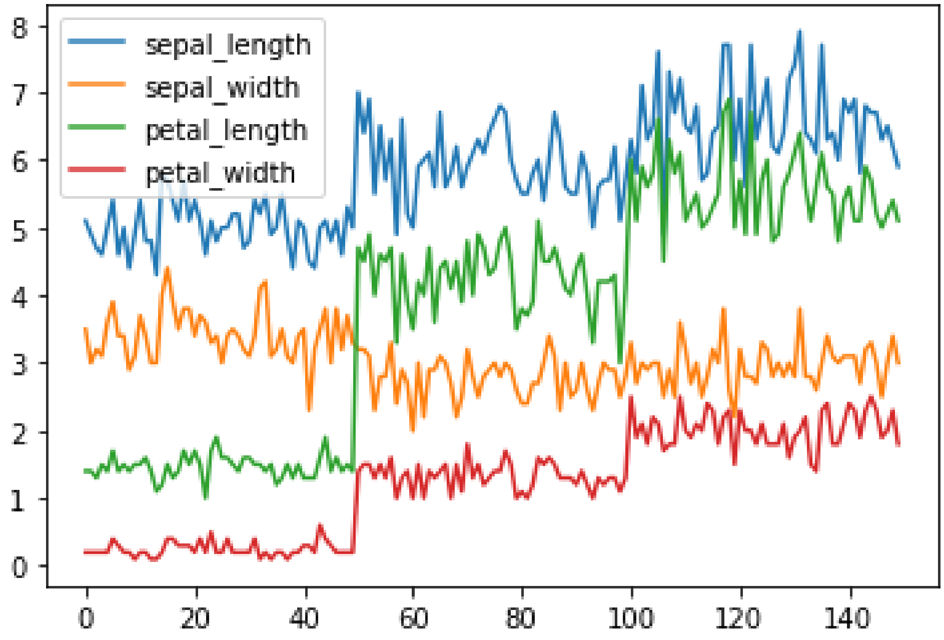
```
[29]: df["petal_width"].plot(kind = 'hist')
```

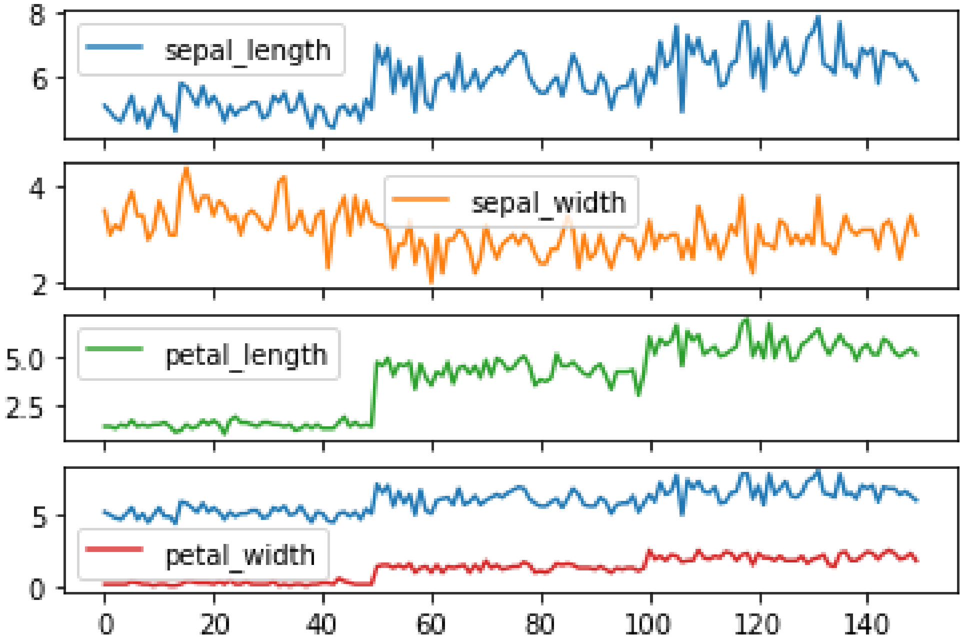
```
[29]: <AxesSubplot:ylabel='Frequency'>
```



[33]:
```
#line plot
df.plot(kind ='line')
df.plot(kind ='line',subplots=True)
```

[33]: `<AxesSubplot:>`



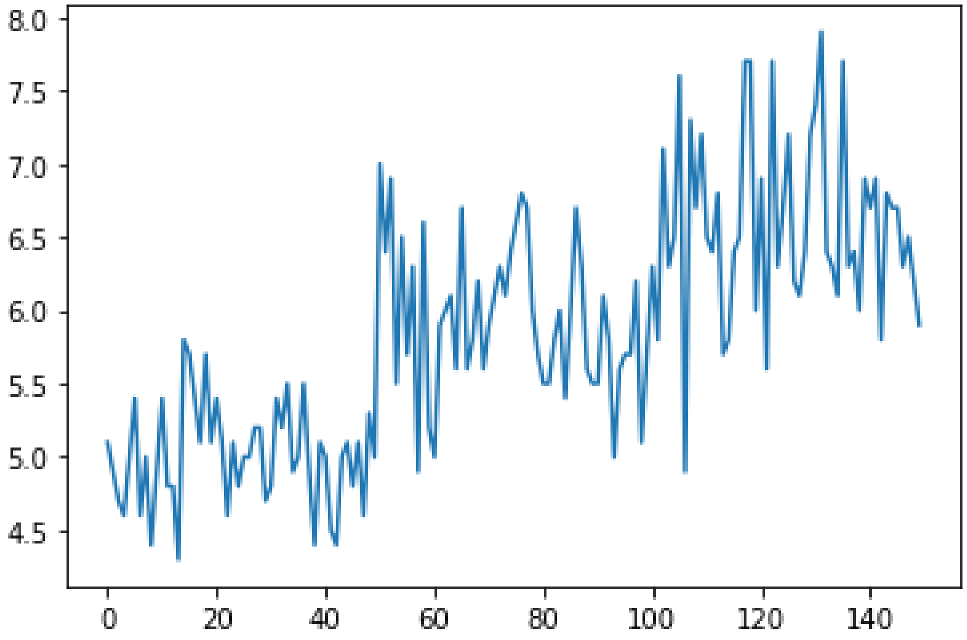


[36]:

```
df["sepal_length"].plot(kind = 'line')
```

[36]:

```
<AxesSubplot:>
```

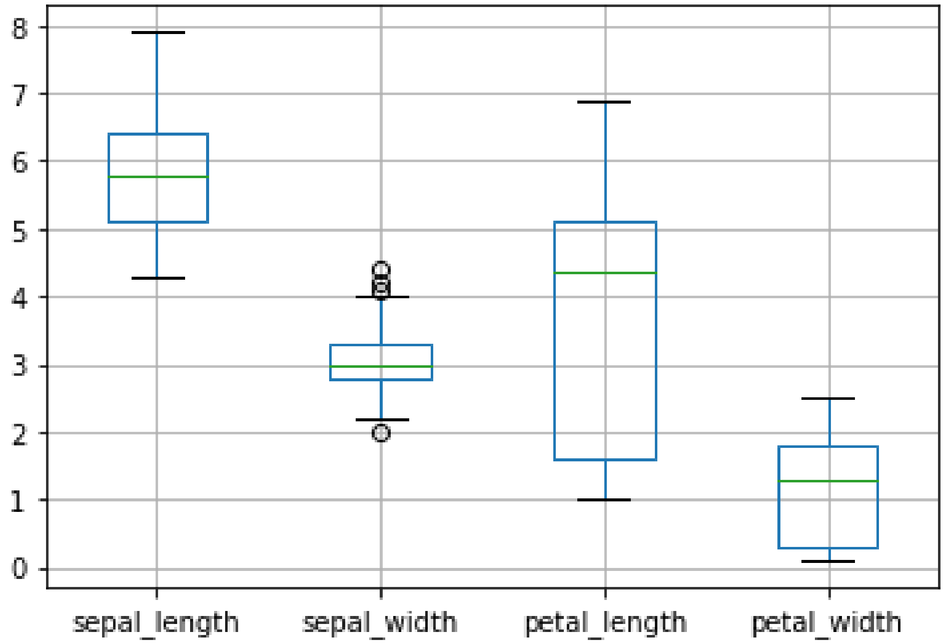


[41]:

```
#box plot
df.boxplot()
```

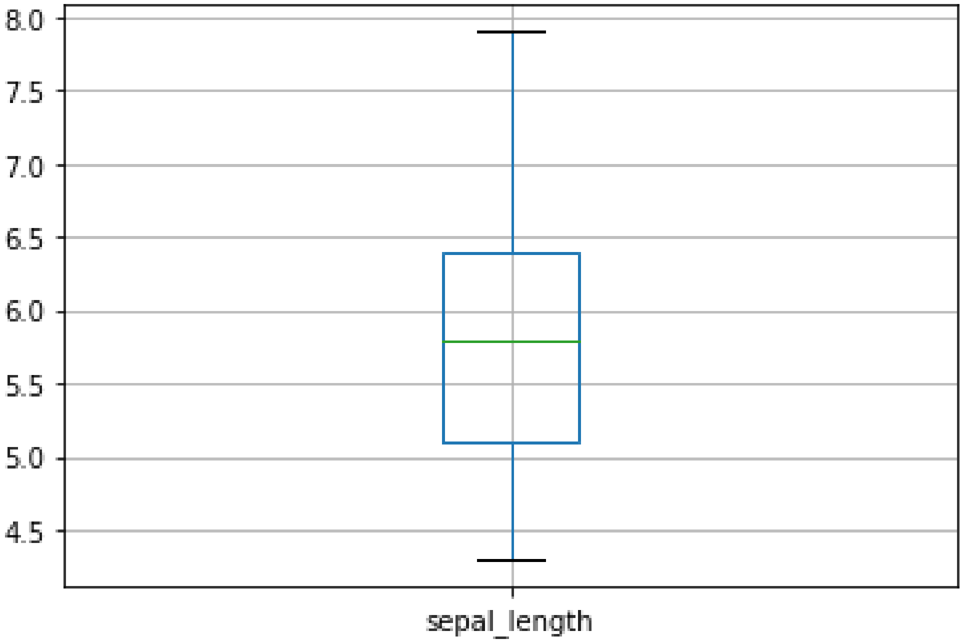
[41]:

```
<AxesSubplot:>
```



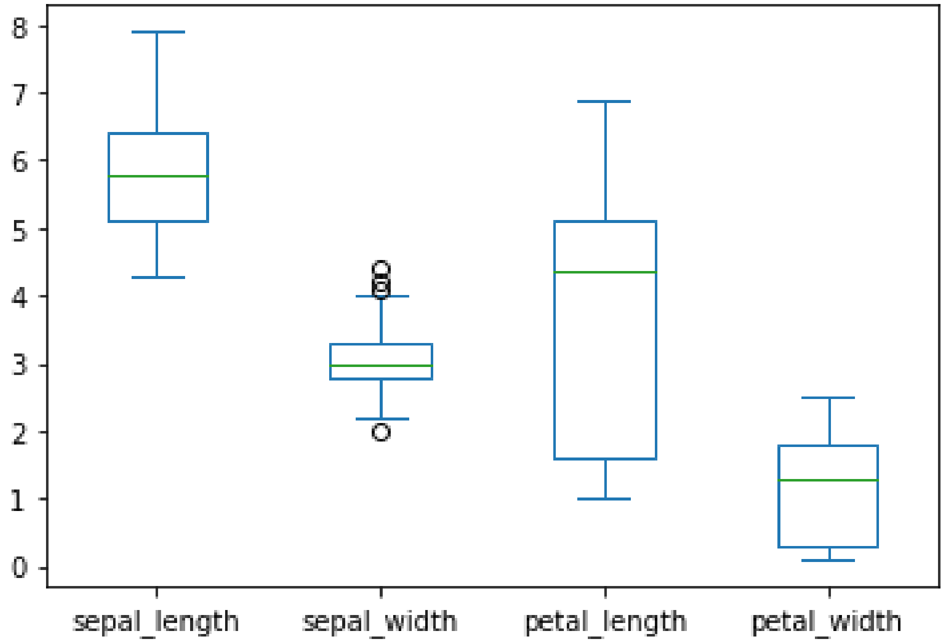
[40]:
```
df.boxplot(["sepal_length"])
```

[40]: <AxesSubplot:>



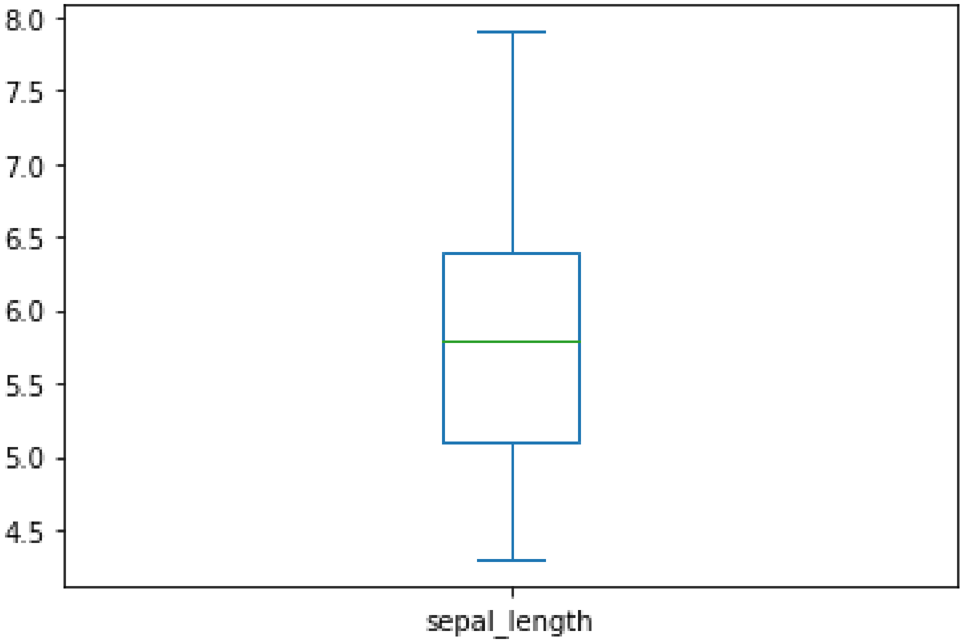
[44]:
```
df.plot(kind='box')
```

[44]:
```
<AxesSubplot:>
```



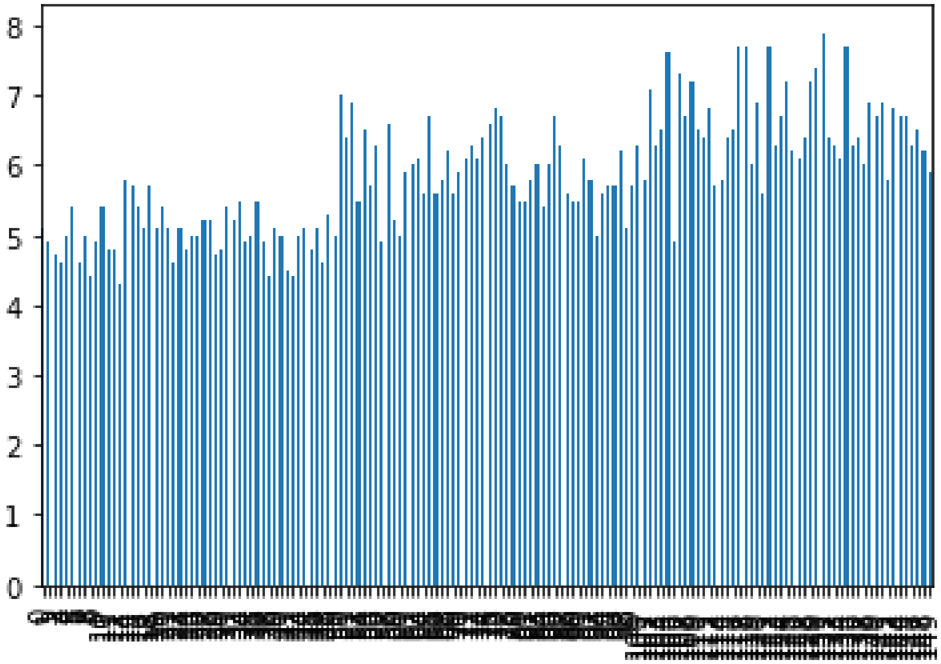
[45]:
```
df["sepal_length"].plot(kind='box')
```

[45]:
```
<AxesSubplot:>
```



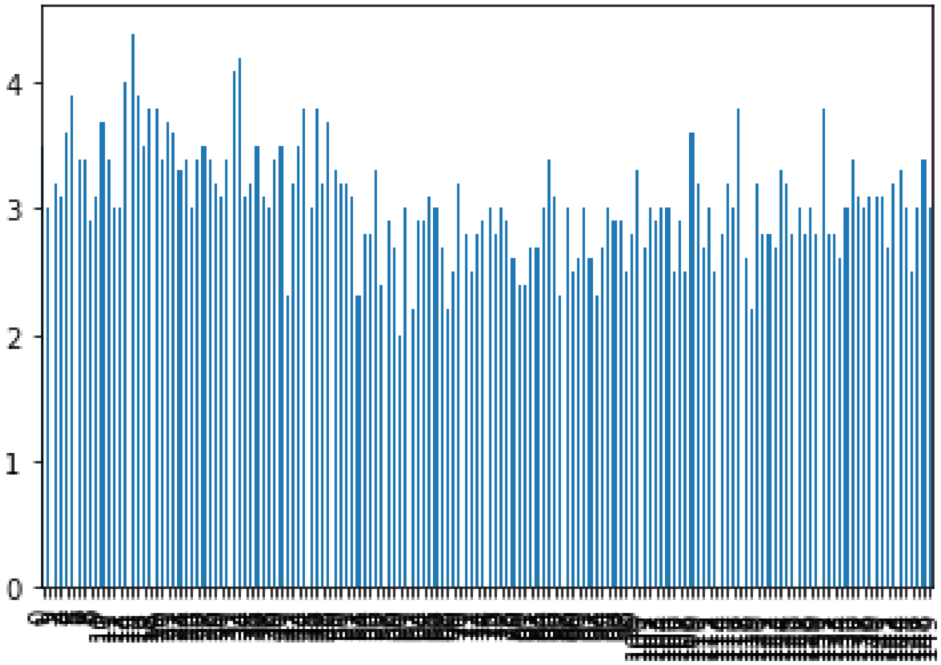
[5]:
```
#bar chart
df["sepal_length"].plot.bar()
```

[5]:
```
<AxesSubplot:>
```



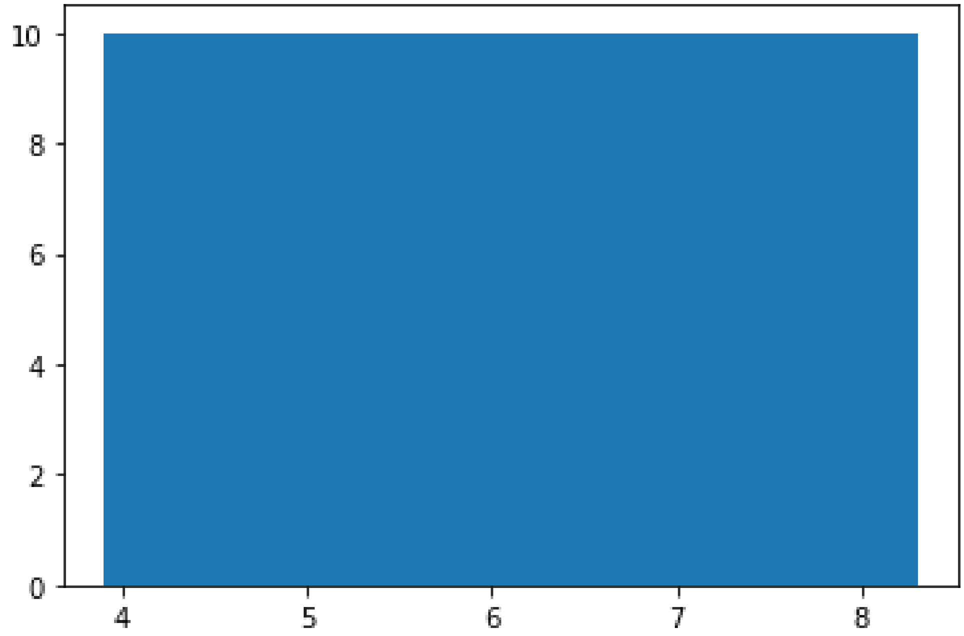
[6]:
```
df["sepal_width"].plot.bar()
```

[6]:
```
<AxesSubplot:>
```



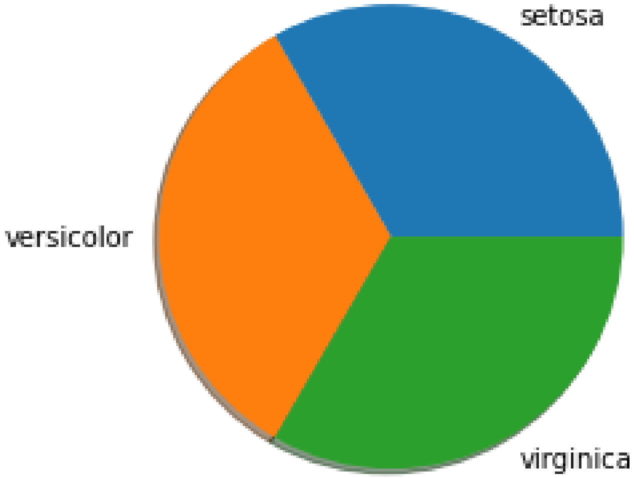
[54]:
```
plt.bar(df["sepal_length"],10)
```

[54]:
```
<BarContainer object of 150 artists>
```



```
[55]: plt.pie(df['species'].
       ↪value_counts(),labels=['setosa','versicolor','virginica'],shadow=True)
```

```
[55]: ([<matplotlib.patches.Wedge at 0x7f75e5a78700>,
        <matplotlib.patches.Wedge at 0x7f75e5a78e50>,
        <matplotlib.patches.Wedge at 0x7f75e5a865e0>],
       [Text(0.5499999702695115, 0.9526279613277875, 'setosa'),
        Text(-1.0999999999999954, -1.0298943258065002e-07, 'versicolor'),
        Text(0.5500001486524352, -0.9526278583383436, 'virginica')])
```

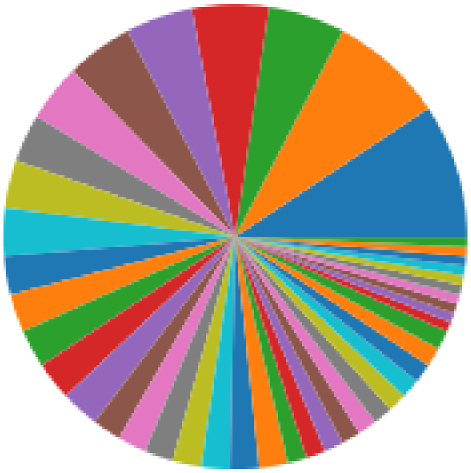


```
[64]: plt.pie(df['species'].value_counts(),shadow=False)
```

```
[64]: ([<matplotlib.patches.Wedge at 0x7f75e2b86580>,
        <matplotlib.patches.Wedge at 0x7f75e2b86a60>,
        <matplotlib.patches.Wedge at 0x7f75e2b86f40>],
       [Text(0.5499999702695115, 0.9526279613277875, ''),
        Text(-1.0999999999999954, -1.0298943258065002e-07, ''),
        Text(0.5500001486524352, -0.9526278583383436, '')])
```

```
[62]: plt.pie(df['petal_length'].value_counts())
```

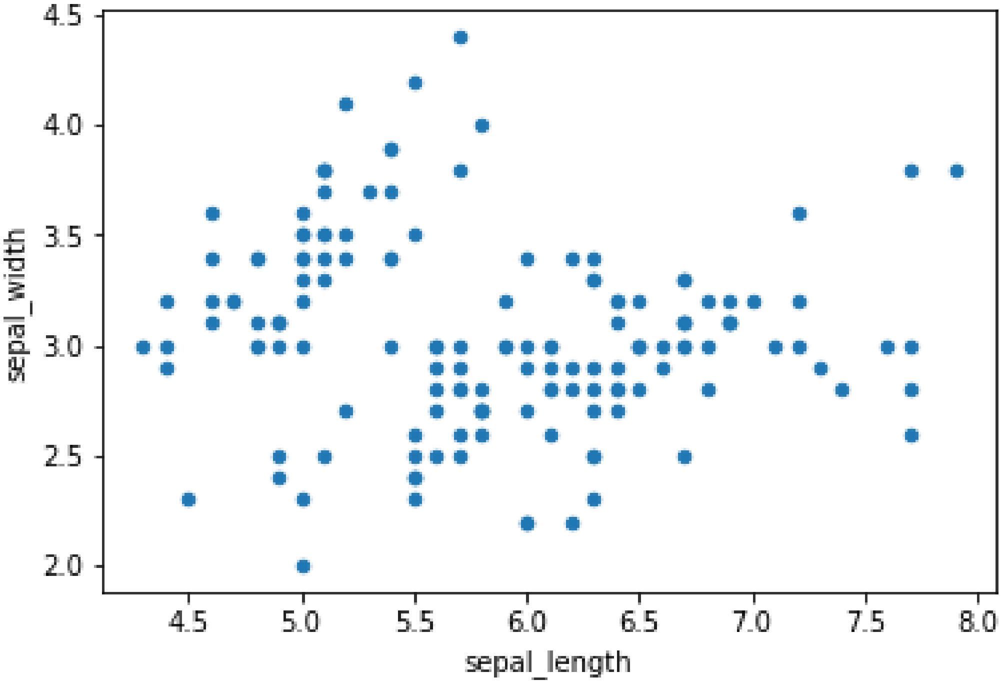
```
[62]: ([<matplotlib.patches.Wedge at 0x7f75e2cc3a60>,
        <matplotlib.patches.Wedge at 0x7f75e2cc3f40>,
        <matplotlib.patches.Wedge at 0x7f75e2cd1460>,
        <matplotlib.patches.Wedge at 0x7f75e2cd1940>,
        <matplotlib.patches.Wedge at 0x7f75e2cd1e20>,
        <matplotlib.patches.Wedge at 0x7f75e2c61340>,
        <matplotlib.patches.Wedge at 0x7f75e2c61820>,
        <matplotlib.patches.Wedge at 0x7f75e2c61d00>,
        <matplotlib.patches.Wedge at 0x7f75e2c6d220>,
        <matplotlib.patches.Wedge at 0x7f75e2c6d700>,
        <matplotlib.patches.Wedge at 0x7f75e2cc3a30>,
        <matplotlib.patches.Wedge at 0x7f75e2c7b0d0>,
        <matplotlib.patches.Wedge at 0x7f75e2c7b5b0>,
        <matplotlib.patches.Wedge at 0x7f75e2c7ba90>,
        <matplotlib.patches.Wedge at 0x7f75e2c7bf70>,
        <matplotlib.patches.Wedge at 0x7f75e2c88490>,
        <matplotlib.patches.Wedge at 0x7f75e2c88970>,
        <matplotlib.patches.Wedge at 0x7f75e2c88e50>,
        <matplotlib.patches.Wedge at 0x7f75e2c17370>,
        <matplotlib.patches.Wedge at 0x7f75e2c17850>,
        <matplotlib.patches.Wedge at 0x7f75e2c17d30>,
        <matplotlib.patches.Wedge at 0x7f75e2c24250>,
        <matplotlib.patches.Wedge at 0x7f75e2c24730>,
        <matplotlib.patches.Wedge at 0x7f75e2c24c10>,
        <matplotlib.patches.Wedge at 0x7f75e2c30130>,
        <matplotlib.patches.Wedge at 0x7f75e2c30610>,
        <matplotlib.patches.Wedge at 0x7f75e2c30af0>,
        <matplotlib.patches.Wedge at 0x7f75e2c30fd0>,
        <matplotlib.patches.Wedge at 0x7f75e2c3e4f0>,
        <matplotlib.patches.Wedge at 0x7f75e2c3e9d0>,
        <matplotlib.patches.Wedge at 0x7f75e2c3eeb0>,
        <matplotlib.patches.Wedge at 0x7f75e2c4a3d0>,
        <matplotlib.patches.Wedge at 0x7f75e2c4a8b0>,
        <matplotlib.patches.Wedge at 0x7f75e2c4ad90>,
        <matplotlib.patches.Wedge at 0x7f75e2bd72b0>,
        <matplotlib.patches.Wedge at 0x7f75e2bd7790>,
        <matplotlib.patches.Wedge at 0x7f75e2bd7c70>,
        <matplotlib.patches.Wedge at 0x7f75e2be4190>,
        <matplotlib.patches.Wedge at 0x7f75e2be4670>,
        <matplotlib.patches.Wedge at 0x7f75e2be4b50>,
        <matplotlib.patches.Wedge at 0x7f75e2bf1070>,
        <matplotlib.patches.Wedge at 0x7f75e2bf1550>,
        <matplotlib.patches.Wedge at 0x7f75e2bf1a30>],
       [Text(1.0530514468883778, 0.3179349779535027, ''),
        Text(0.7360436695459462, 0.817459305728021, ''),
        Text(0.33991869870988073, 1.0461621663333946, ''),
        Text(-0.023036665647143535, 1.0997587517432457, ''),
        Text(-0.3617533198417098, 1.0388140043258474, ''),
        Text(-0.6465637858537404, 0.8899186877588754, ''),
        Text(-0.8620628079629176, 0.6832625521182104, ''),
        Text(-0.995309762316952, 0.4683572109368796, ''),
        Text(-1.070936798373572, 0.25118593489557295, ''),
        Text(-1.0997587525520631, 0.023036627034576564, ''),
        Text(-1.0845956335925047, -0.18344566386827935, ''),
        Text(-1.0388139916246755, -0.36175335631450445, ''),
        Text(-0.9639373229924024, -0.5299290870883018, ''),
        Text(-0.8620627679806308, -0.6832626025632996, ''),
        Text(-0.7360436217109051, -0.8174593487988833, ''),
        Text(-0.6087306528413958, -0.9162133988822081, ''),
        Text(-0.48909864467532527, -0.9852829622888849, ''),
        Text(-0.36175325905371764, -1.0388140254944644, ''),
        Text(-0.22870280869061121, -1.075962371691978, ''),
        Text(-0.09204557857504166, -1.0961421492966987, ''),
        Text(0.046063265235212945, -1.099035111175194, ''),
        Text(0.18344566386827924, -1.0845956335925047, ''),
        Text(0.29581184376028896, -1.059478812006705, ''),
        Text(0.38342929412504956, -1.0310101727949956, ''),
        Text(0.46835726335491756, -0.9953097376508945, ''),
        Text(0.5500000427375765, -0.95262791948833, ''),
        Text(0.6277849668081942, -0.9032641006093592, ''),
        Text(0.7011664301863999, -0.8475645327523212, ''),
        Text(0.769629714606191, -0.7859199083845587, ''),
        Text(0.8326945992949211, -0.7187626202753389, ''),
        Text(0.8899187293771469, -0.6465637285711012, ''),
        Text(0.9409007186763939, -0.5698296566468314, ''),
        Text(0.9852829651509297, -0.4890986389097725, ''),
        Text(1.0140494925724515, -0.4262670836618211, ''),
        Text(1.0310102120597853, -0.3834291885452085, ''),
        Text(1.0461621892079598, -0.33991862830920133, ''),
        Text(1.0594788422990937, -0.2958117352651384, ''),
        Text(1.0709368093975045, -0.25118588789478863, ''),
        Text(1.080515989334485, -0.2061193751021953 8, ''),
        Text(1.0881995769725843, -0.16069125886833013, ''),
        Text(1.0939740926875325, -0.11498123554950176, ''),
        Text(1.0978294060160814, -0.06906949606286578, ''),
        Text(1.0997587534282804, -0.02303658520429573, '')])
```



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

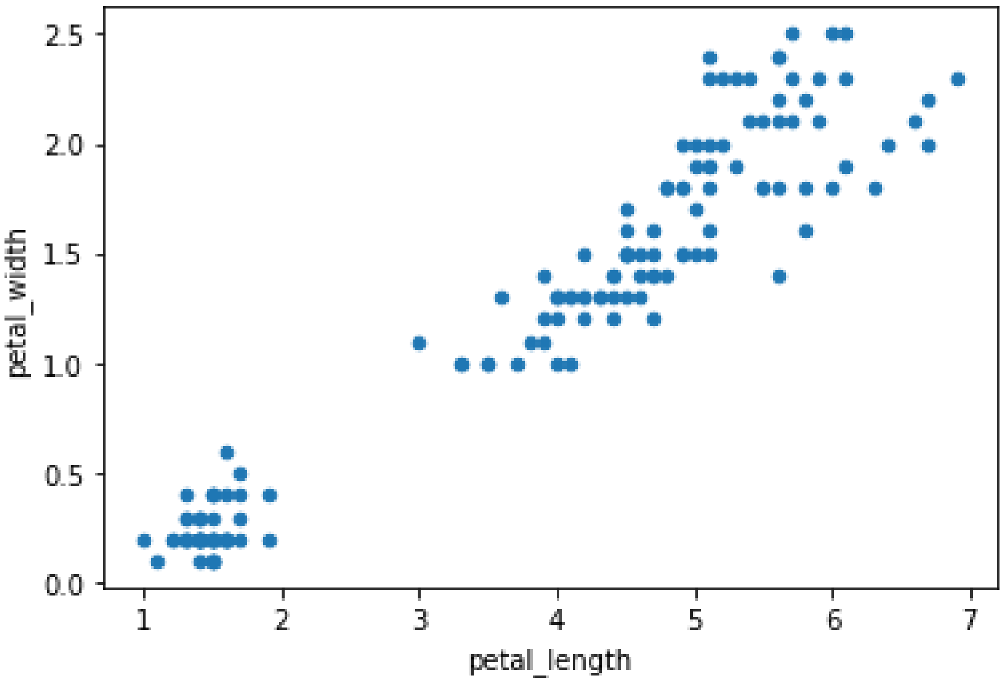
```
[66]: #scatter plot
      df.plot(kind='scatter',x='sepal_length',y='sepal_width')
```

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



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

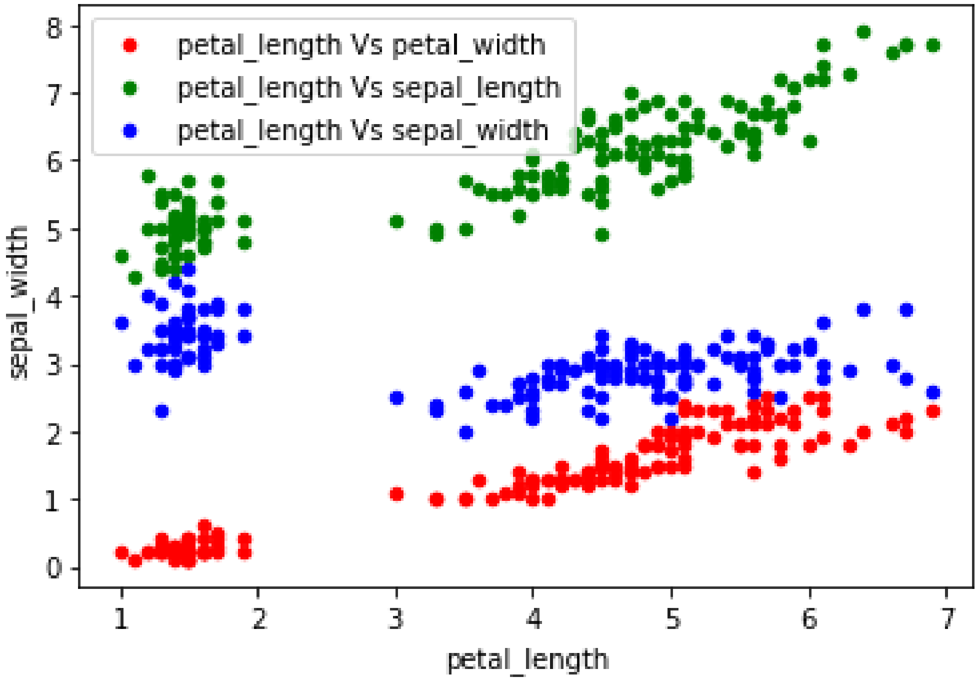
```
[67]: <AxesSubplot:xlabel='petal_length', ylabel='petal_width'>
```



[68]:

```
#scatter multiple
ax1 = df.
 ↪plot(kind='scatter',x='petal_length',y='petal_width',label='petal_length Vs␣
 ↪petal_width',color='r')
ax2 = df.
 ↪plot(kind='scatter',x='petal_length',y='sepal_length',label='petal_length Vs␣
 ↪sepal_length',color='g',ax=ax1)
ax3 = df.
 ↪plot(kind='scatter',x='petal_length',y='sepal_width',label='petal_length Vs␣
 ↪sepal_width', color='b',ax=ax2)
print(ax1 == ax2 == ax3)
```

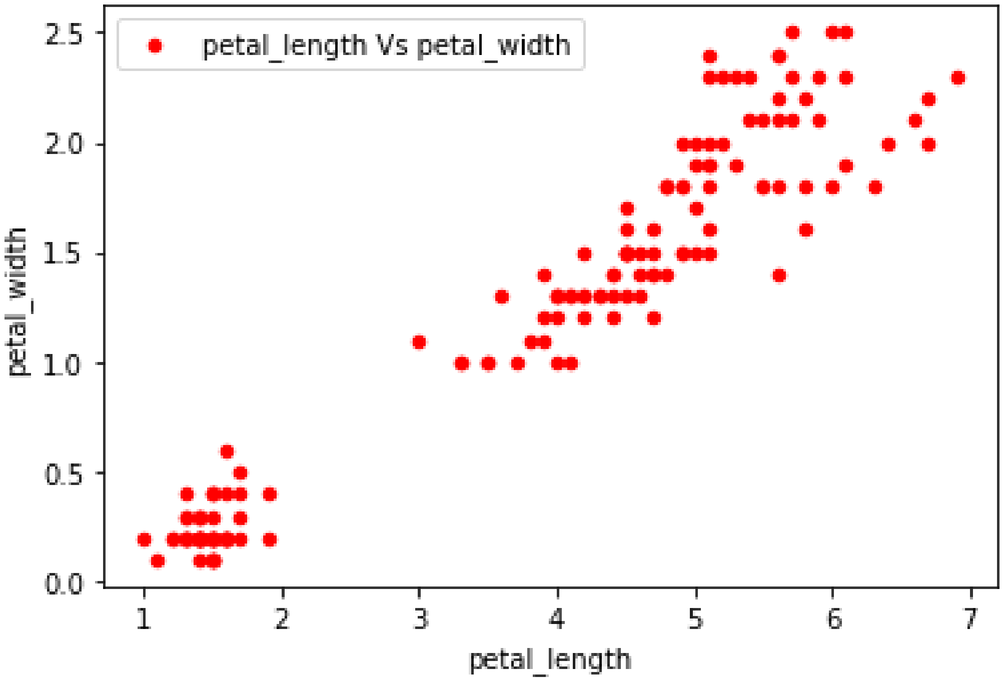
```
True
```



[71]:

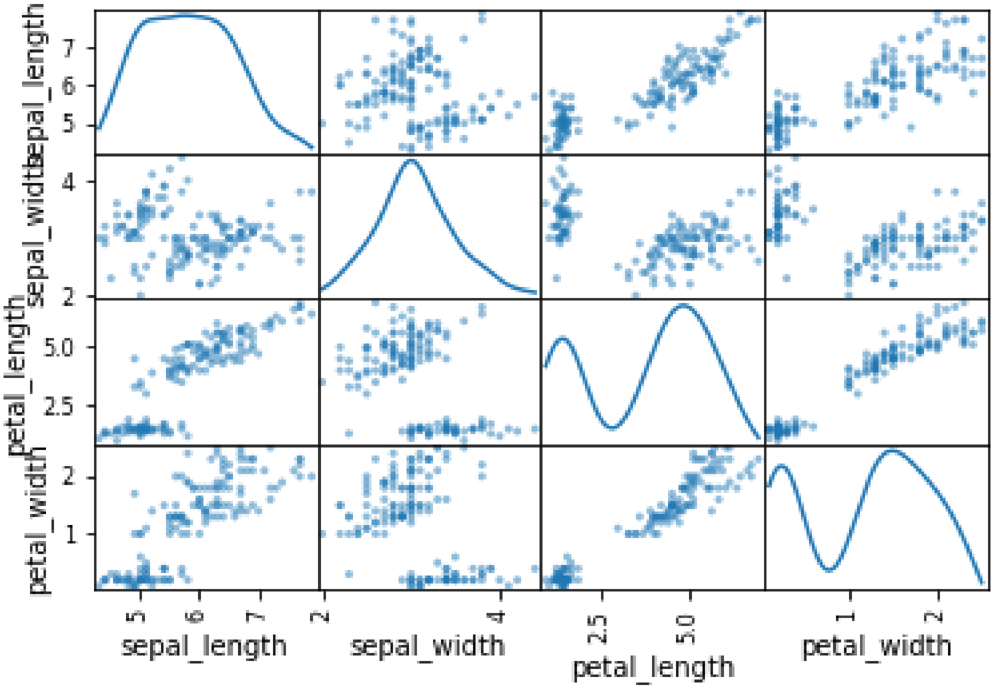
```
ax1 = df.
 ↪plot(kind='scatter',x='petal_length',y='petal_width',label='petal_length Vs␣
 ↪petal_width',color='r')
print(ax1)
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



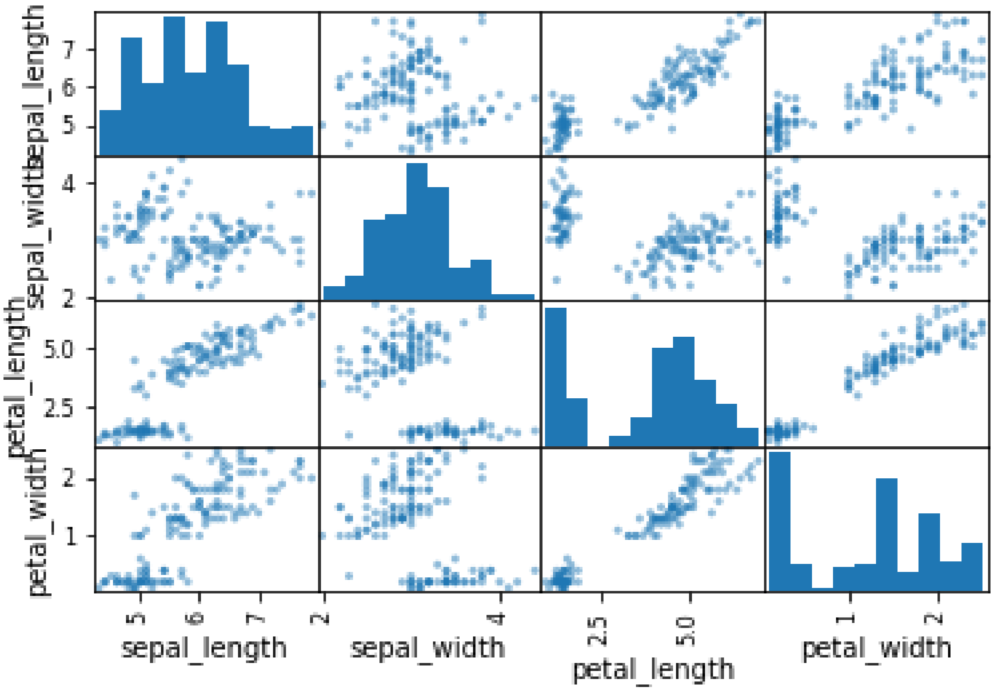
```
[73]: #scatter matrix
      from pandas.plotting import scatter_matrix
      scatter_matrix(df, alpha=0.5, diagonal='kde')
```

```
[73]: 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)
```



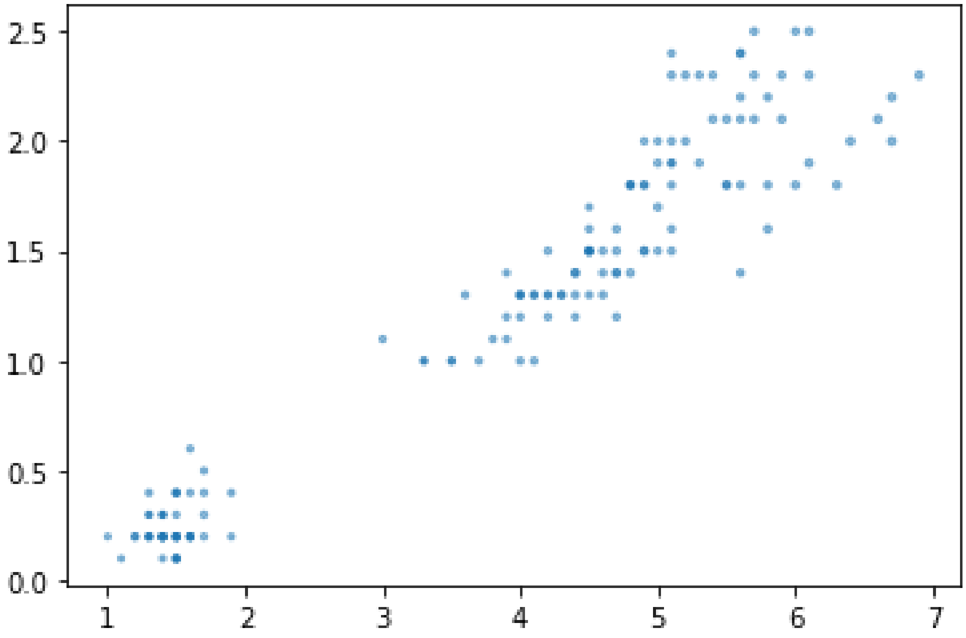
```
[74]: pd.plotting.scatter_matrix(df)
```

```
[74]: 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)
```



[76]:
```
#bubble plot
plt.scatter('petal_length','petal_width',s='sepal_length',alpha=0.5,data=df)
```

[76]:
```
<matplotlib.collections.PathCollection at 0x7f75ddf231f0>
```

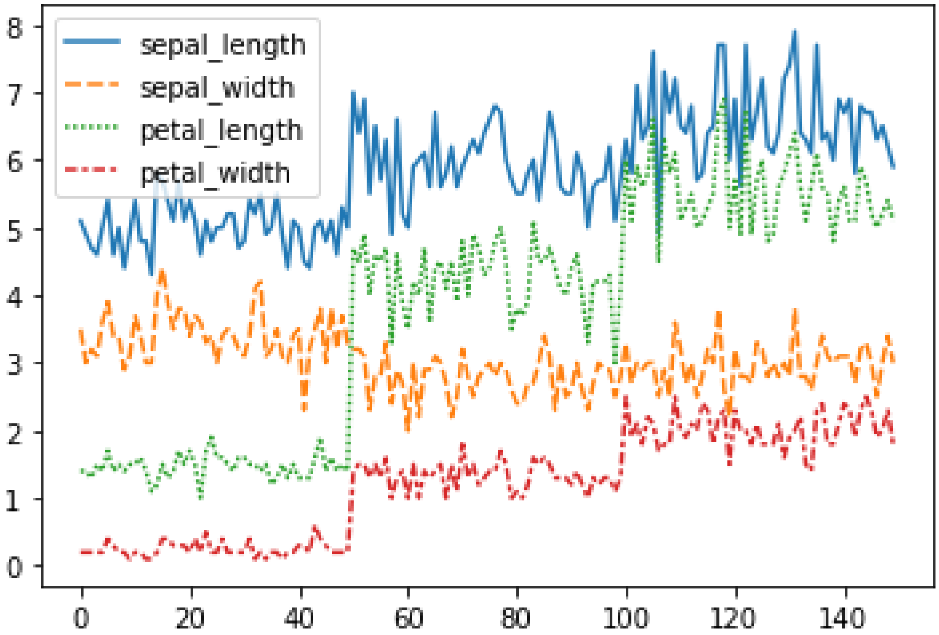


c)Write a program to draw univariate and multivariate visualization plots with seaborn(line plot, pairplot,jointplot,heatmap) for iris dataset

```
[77]: import seaborn as sns
```

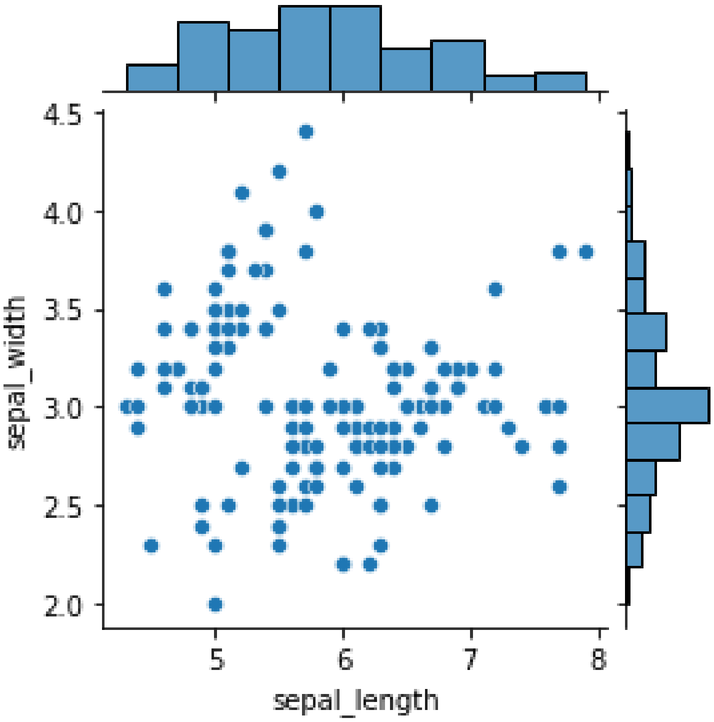
```
[78]: #line plot
      sns.lineplot(data=df)
```

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



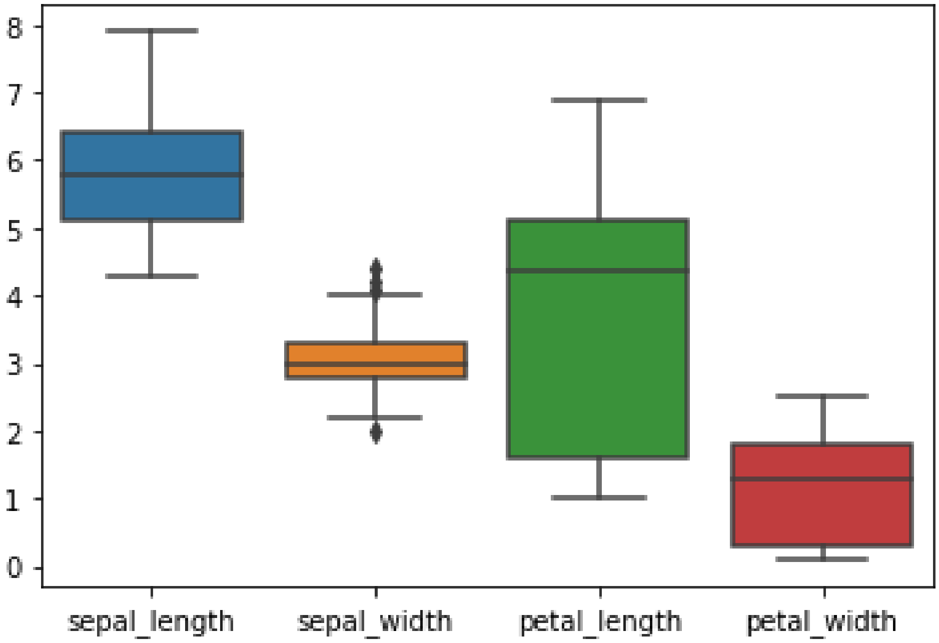
```
[79]: #joint plot
      sns.jointplot(x='sepal_length',y='sepal_width',data=df,height=4)
```

```
[79]: <seaborn.axisgrid.JointGrid at 0x7f75dddcefa0>
```



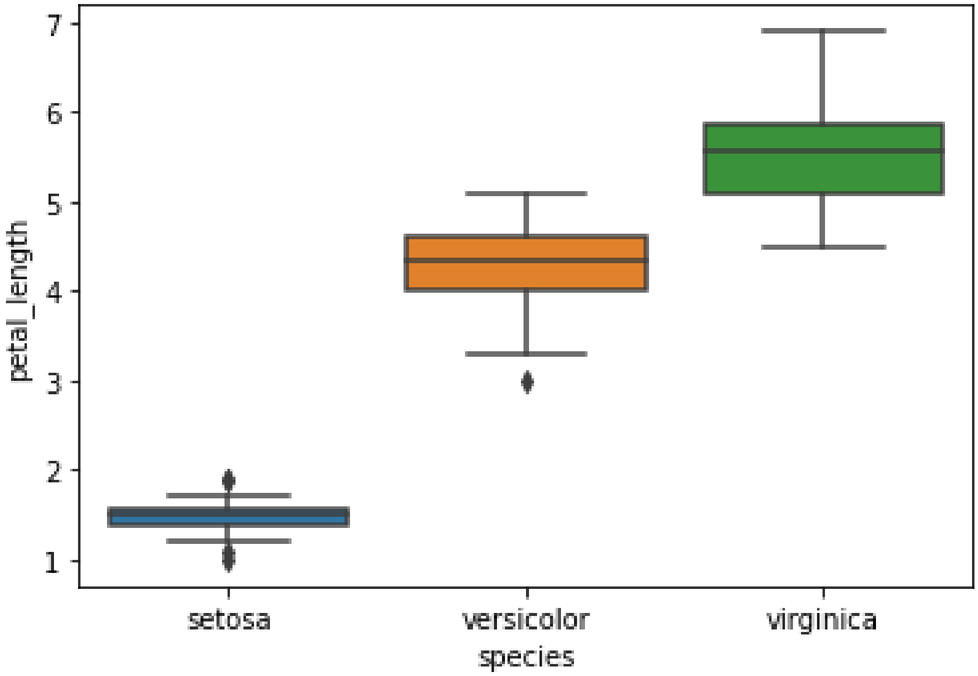
[81]:
```
#box plot
sns.boxplot(data=df)
```

[81]:
```
<AxesSubplot:>
```



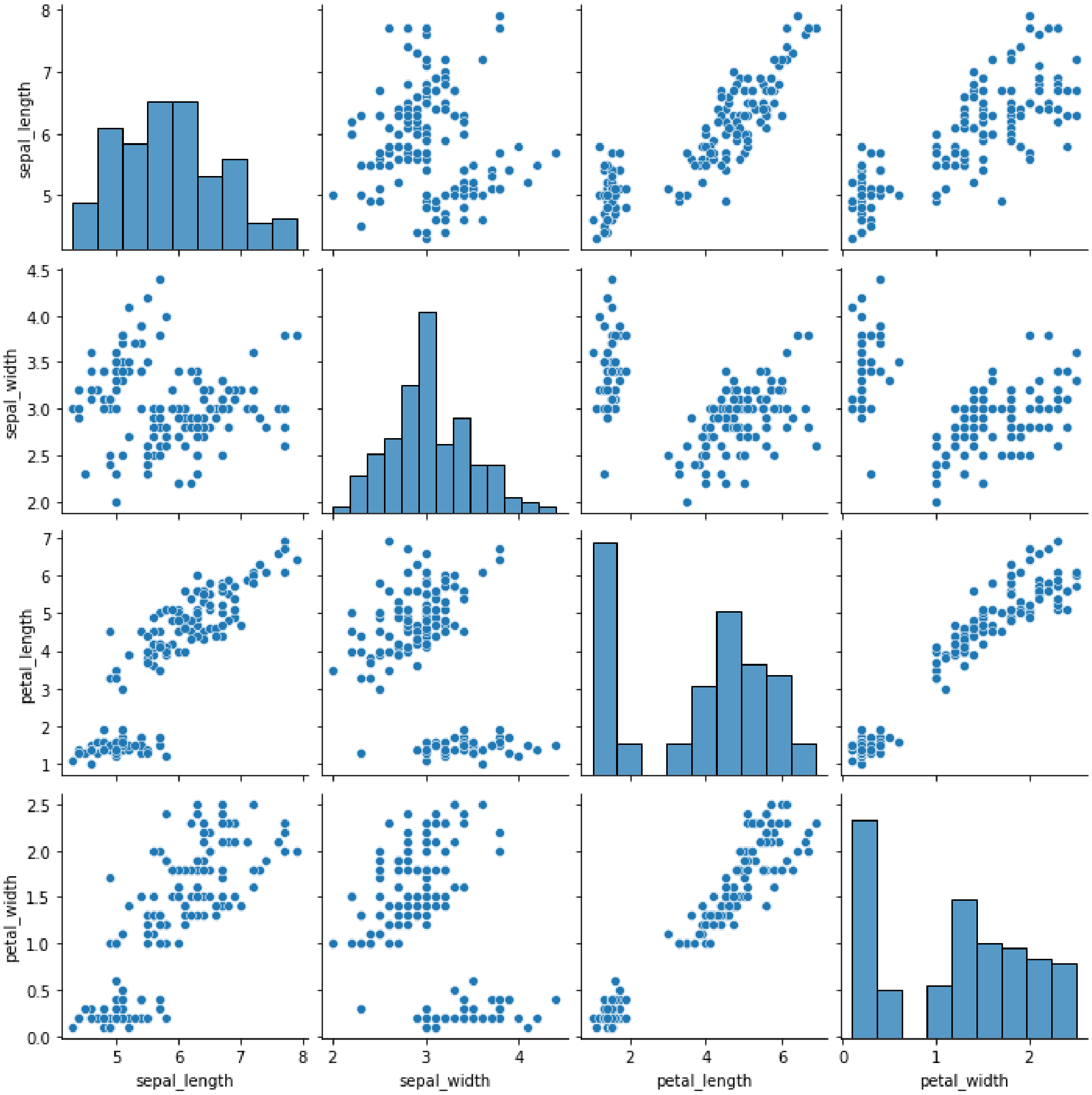
[82]:
```
sns.boxplot(x='species',y='petal_length',data=df)
```

[82]:
```
<AxesSubplot:xlabel='species', ylabel='petal_length'>
```



[83]:
```
#pair plot
sns.pairplot(data=df,kind='scatter')
```

[83]:
```
<seaborn.axisgrid.PairGrid at 0x7f75dd845100>
```

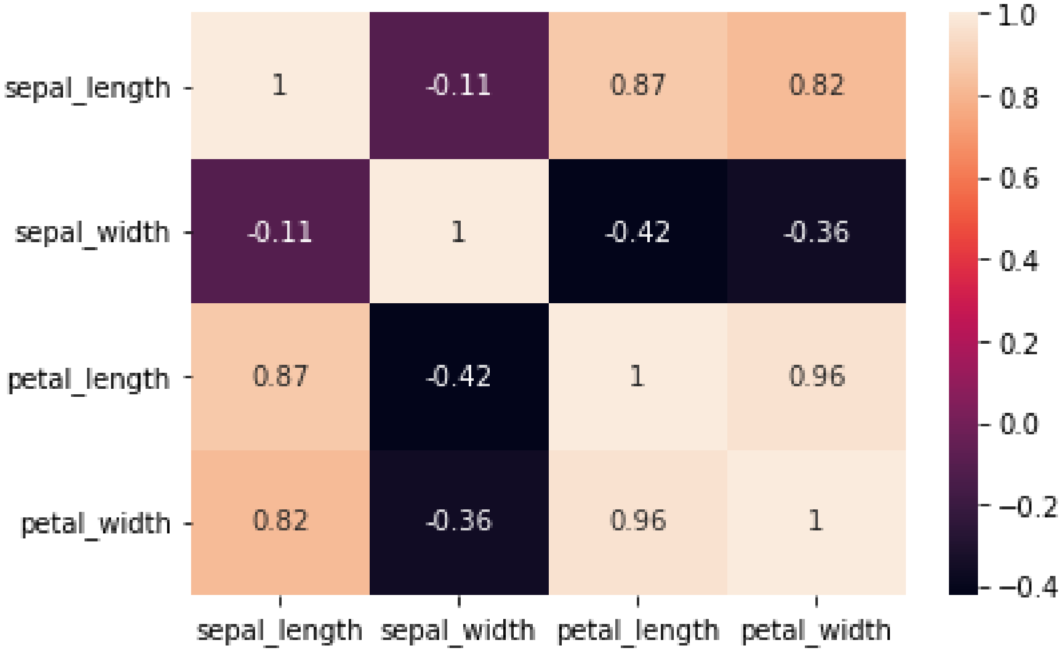


[84]:

```
#heat map
sns.heatmap(df.corr(),annot=True)
```

[84]:

```
<AxesSubplot:>
```


sepal_length
sepal_width
petal_length
petal_width
1
-0.11
0.87
0.82
-0.11
1
-0.42
-0.36
0.87
-0.42
1
0.96
0.82
-0.36
0.96
1
sepal_length
sepal_width
petal_length
petal_width
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

[ ]: