Data visualization using pandas

In [2]:

```
## Import libraries
import numpy as np
import pandas as pd
from numpy.random import randn, randint, uniform, sample
```

In [3]:

```
pd.DataFrame(randn(100))
```

Out[3]:

	0
0	-1.362824
1	-1.303100
2	-0.078055
3	-0.368322
4	0.894979
...	...
95	0.596446
96	-1.272193
97	-0.207650
98	-0.857266
99	-0.080833

100 rows × 1 columns

In [5]:

```
df=pd.DataFrame(randn(100), index=pd.date_range('01-01-2021', periods=100), columns=['va
lue'])
df.tail()
```

Out[5]:

	value
2021-04-06	0.051694
2021-04-07	0.736291
2021-04-08	-0.723140
2021-04-09	-0.295126
2021-04-10	-0.419689

In [6]:

```
type(df)
```

Out[6]:

```
pandas.core.frame.DataFrame
```

In [9]:

```
ds=pd.Series(randn(100), index=pd.date_range('01-01-2021', periods=100))
```

```
ds.head()
```

Out[9]:

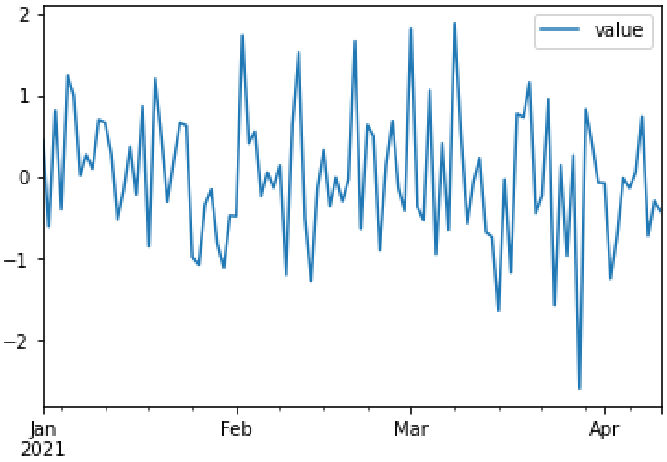
```
2021-01-01    0.487739
2021-01-02   -0.420996
2021-01-03    0.554809
2021-01-04   -1.263474
2021-01-05    0.908801
Freq: D, dtype: float64
```

In [10]:

```
df.plot()
```

Out[10]:

```
<AxesSubplot:>
```

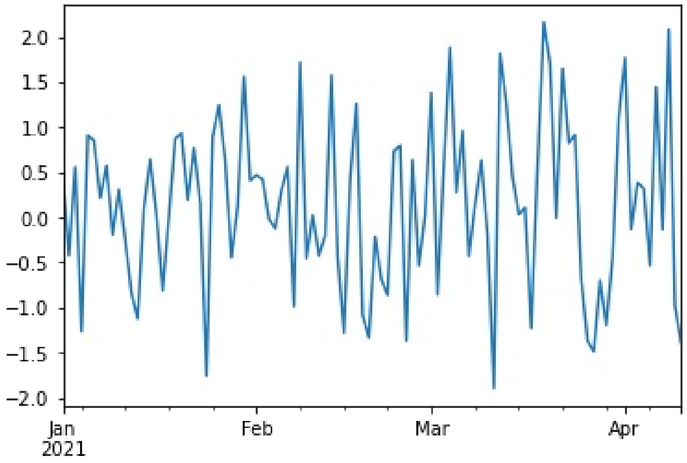


In [12]:

```
ds.plot()
```

Out[12]:

```
<AxesSubplot:>
```



In [13]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [16]:

```
x = np.linspace(0, 10, 50)
y = np.sin(x)
```

In [17]:

```
x
```

Out[17]:

```
array([ 0.        ,  0.20408163,  0.40816327,  0.6122449 ,  0.81632653,
        1.02040816,  1.2244898 ,  1.42857143,  1.63265306,  1.83673469,
        2.04081633,  2.24489796,  2.44897959,  2.65306122,  2.85714286,
        3.06122449,  3.26530612,  3.46938776,  3.67346939,  3.87755102,
        4.08163265,  4.28571429,  4.48979592,  4.69387755,  4.89795918,
        5.10204082,  5.30612245,  5.51020408,  5.71428571,  5.91836735,
        6.12244898,  6.32653061,  6.53061224,  6.73469388,  6.93877551,
        7.14285714,  7.34693878,  7.55102041,  7.75510204,  7.95918367,
        8.16326531,  8.36734694,  8.57142857,  8.7755102 ,  8.97959184,
        9.18367347,  9.3877551 ,  9.59183673,  9.79591837, 10.        ])
```

In [18]:

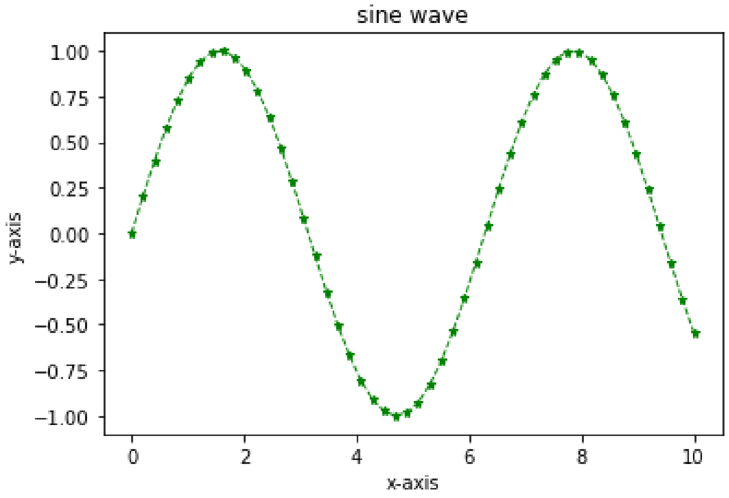
```
y
```

Out[18]:

```
array([ 0.        ,  0.20266794,  0.39692415,  0.57470604,  0.72863478,
        0.85232157,  0.94063279,  0.98990308,  0.99808748,  0.96484631,
        0.89155923,  0.78126802,  0.63855032,  0.46932961,  0.2806294 ,
        0.08028167, -0.12339814, -0.32195632, -0.50715171, -0.67129779,
       -0.80758169, -0.91034694, -0.97532829, -0.99982867, -0.9828312 ,
       -0.92504137, -0.82885774, -0.6982724 , -0.53870529, -0.35677924,
       -0.16004509,  0.04333173,  0.24491007,  0.43632343,  0.6096272 ,
        0.75762842,  0.8741843 ,  0.9544572 ,  0.99511539,  0.99447137,
        0.95255185,  0.8710967 ,  0.75348673,  0.60460332,  0.43062587,
        0.23877532,  0.0370144 , -0.16628279, -0.36267843, -0.54402111])
```

In [27]:

```
## Line plot
#plt.plot(x,y,'g*')
plt.plot(x,y,color='green', marker='*', linestyle='dashed',linewidth=1, markersize=5)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('sine wave')
plt.show()
```



In [31]:

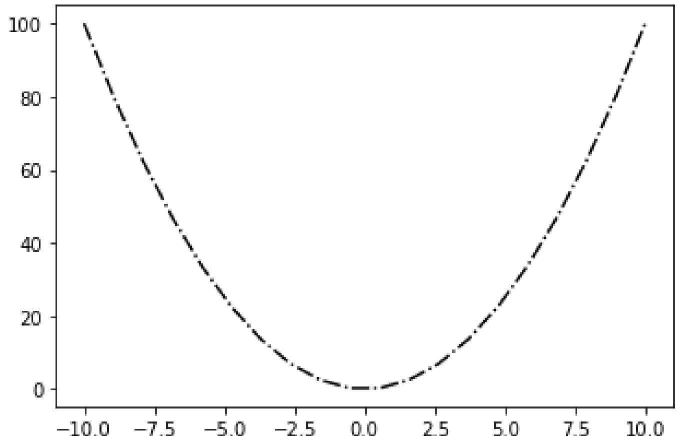
```
x=np.linspace(-10,10,20)
y=x**2
x
```

Out[31]:

```
array([-10.        ,  -8.94736842,  -7.89473684,  -6.84210526,
        -5.78947368,  -4.73684211,  -3.68421053,  -2.63157895,
        -1.57894737,  -0.52631579,   0.52631579,   1.57894737,
         2.63157895,   3.68421053,   4.73684211,   5.78947368,
         6.84210526,   7.89473684,   8.94736842,  10.        ])
```

In [34]:

```
plt.plot(x,y,'k-.')
plt.show()
```



In [35]:

```
import seaborn as sns
```

In [36]:

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

Out[36]:

	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

In [37]:

```
iris.shape
```

Out[37]:

```
(150, 5)
```

In [39]:

```
iris['species'].unique()
```

Out[39]:

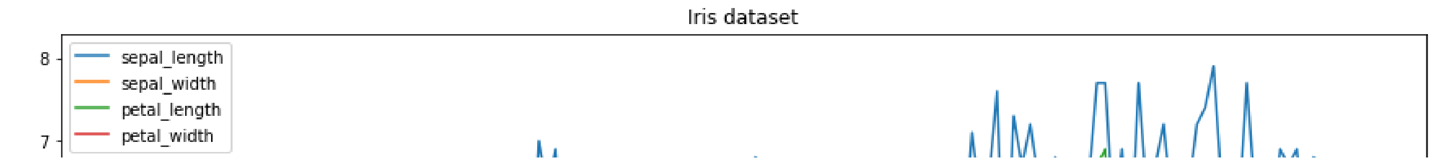
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

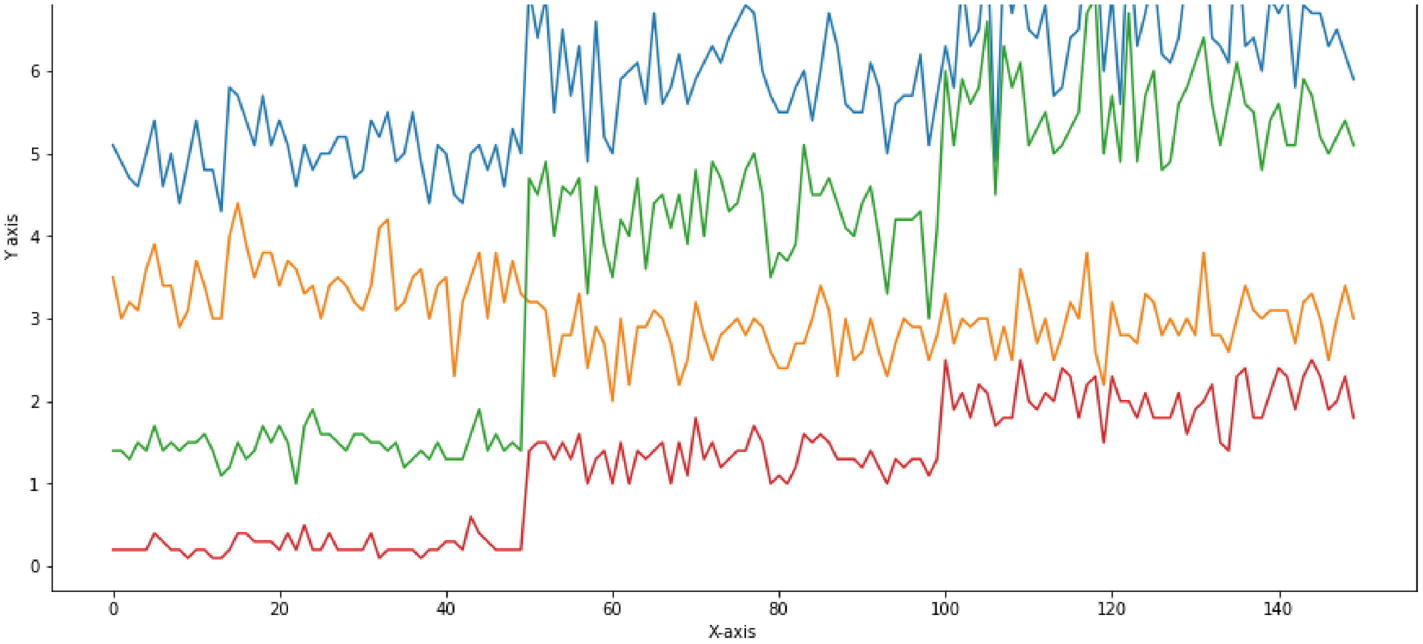
In [42]:

```
ax=iris.plot(figsize=(15,8), title='Iris dataset')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y axis')
```

Out[42]:

```
Text(0, 0.5, 'Y axis')
```





In [43]:

```
df=iris.drop(['species'], axis=1)
df
```

Out[43]:

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

150 rows × 4 columns

In [45]:

```
df.iloc[0]   ### data from 0th index
```

Out[45]:

```
sepal_length    5.1
sepal_width     3.5
petal_length    1.4
petal_width     0.2
Name: 0, dtype: float64
```

In [72]:

```
x=[1,2,3,4,5,6]
y=[1,4,6,8,10,12]
```

In [73]:

```
x2=np.arange(0,4,0.5)
```

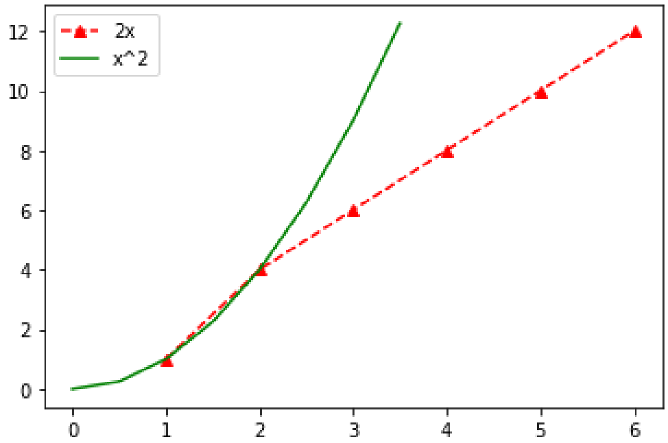
```
x2**2
```

Out[73]:

```
array([ 0.  ,  0.25,  1.  ,  2.25,  4.  ,  6.25,  9.  , 12.25])
```

In [75]:

```
plt.plot(x,y,'r^--', label='2x')
plt.plot(x2, x2**2, 'g', label='x^2')
plt.legend()
plt.show()
```



Bar charts

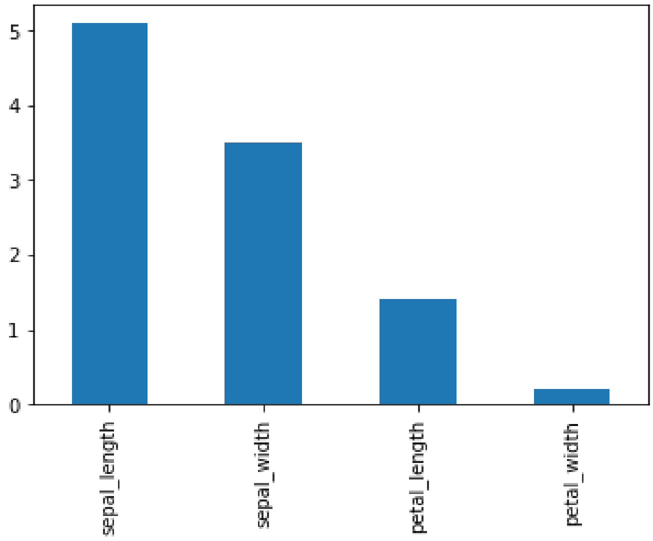
A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

In [47]:

```
df.iloc[0].plot(kind='bar')
```

Out[47]:

```
<AxesSubplot:>
```

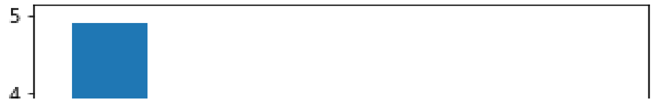


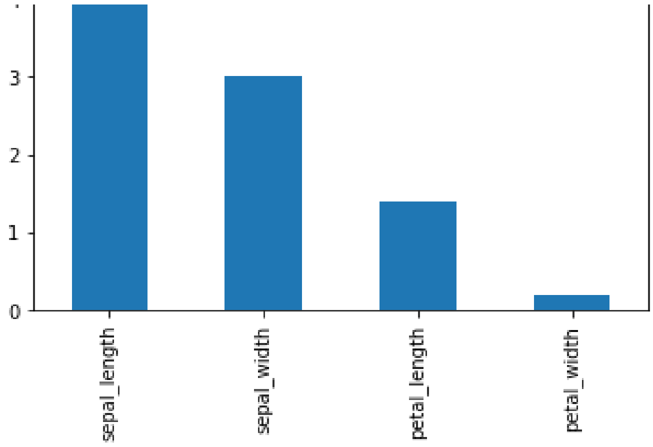
In [48]:

```
df.iloc[1].plot.bar()
```

Out[48]:

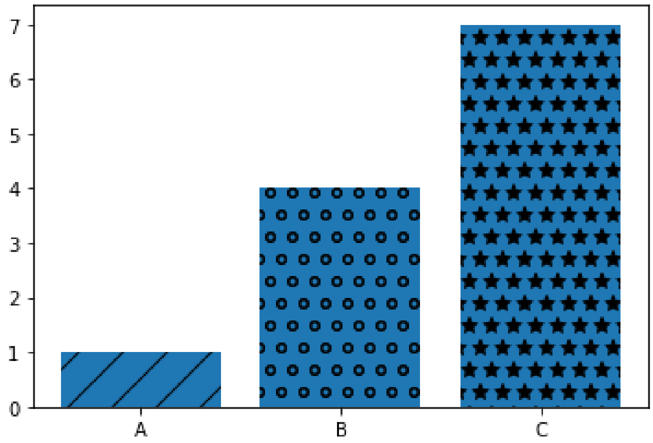
```
<AxesSubplot:>
```





In [80]:

```
labels=['A', 'B', 'C']
values=[1,4,7]
bars=plt.bar(labels,values)
patterns=['/', 'O', '*']
bars[0].set_hatch('/')
bars[1].set_hatch('o')
bars[2].set_hatch('*')
plt.show()
```



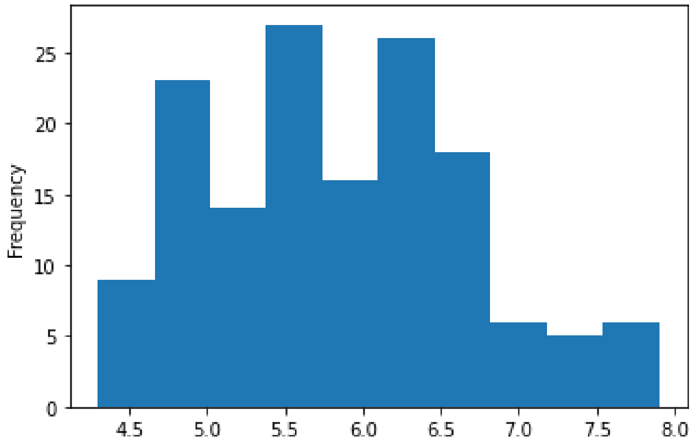
Histogram

A histogram is a graph showing frequency distributions.

It is a graph showing the number of observations within each given interval

In [52]:

```
iris['sepal_length'].plot.hist()
plt.show()
```

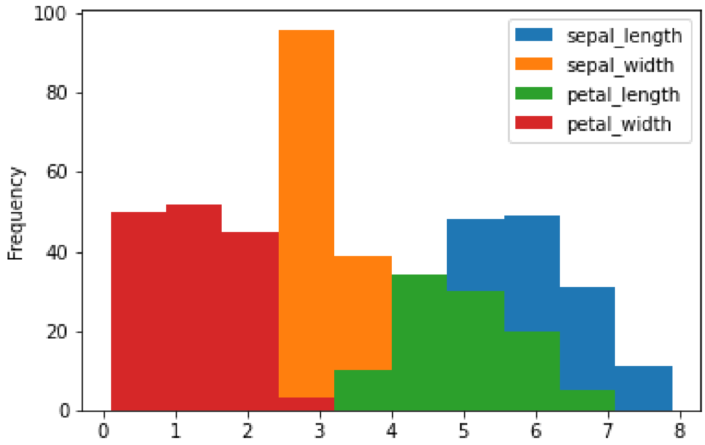


In [56]:

```
#iris.plot.hist()
iris.plot(kind='hist')
```

Out[56]:

```
<AxesSubplot:ylabel='Frequency'>
```

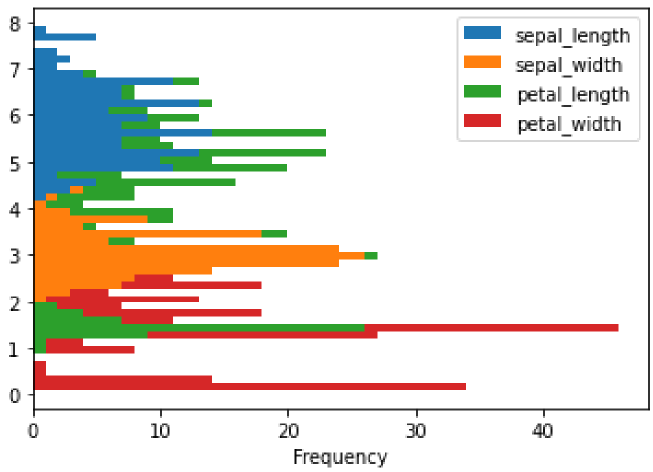


In [60]:

```
iris.plot(kind='hist', stacked=True, bins=50, orientation='horizontal')
```

Out[60]:

```
<AxesSubplot:xlabel='Frequency'>
```



In [62]:

```
df=iris.drop(['species'], axis=1)
df.diff().head()
```

Out[62]:

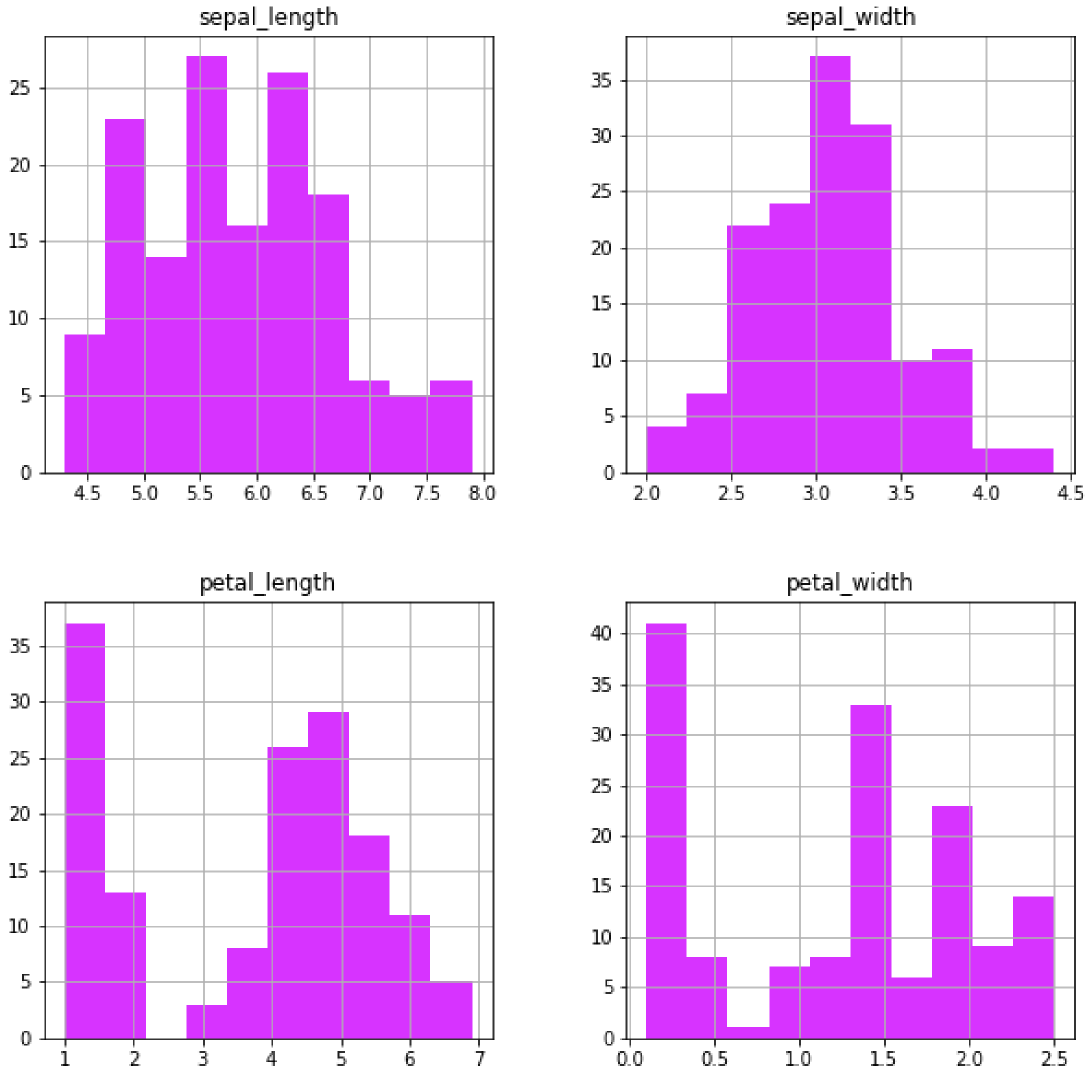
	sepal_length	sepal_width	petal_length	petal_width
0	NaN	NaN	NaN	NaN
1	-0.2	-0.5	0.0	0.0
2	-0.2	0.2	-0.1	0.0
3	-0.1	-0.1	0.2	0.0
4	0.4	0.5	-0.1	0.0

In [66]:

```
df.hist(color='#D733FF', figsize=(10,10))
```

Out[66]:

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



Scatter

The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis

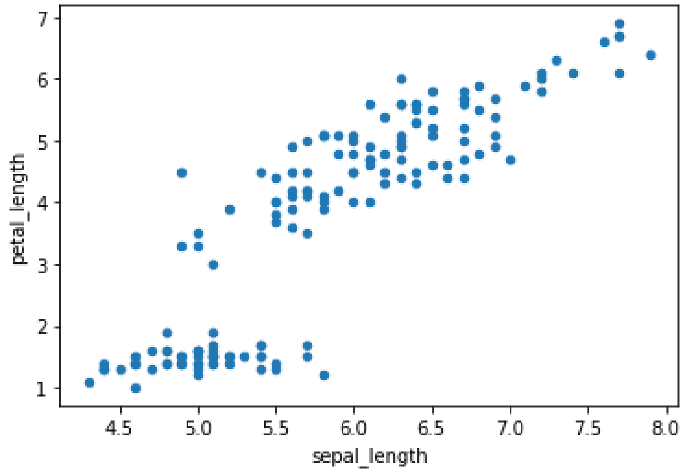
In [67]:

```
## scatter plot

iris.plot.scatter(x='sepal_length', y='petal_length')
```

Out[67]:

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



In [ ]:

```
```