Data Visualisation with Pandas

In [83]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [84]:

```
from numpy.random import randn, randint, uniform, sample
```

In [85]:

```
randn(37)
```

Out[85]:

```
array([-0.55690622, -0.03751876,  0.82068771,  1.29780433, -0.61895997,
       -1.26962676, -0.80350589, -1.82609206,  0.35996217,  1.43234392,
       -0.05136391,  0.39842093, -1.52355509,  0.09396123,  1.21326184,
        0.5035854 , -2.0654554 , -1.2865167 , -0.1451684 , -0.47946109,
       -1.69180857,  0.23727778, -0.45285492, -0.38054141, -2.69618335,
        0.59742896, -1.20361583,  0.06159641, -0.28296104, -0.47854897,
        0.94106431,  1.25118202,  0.09786005, -1.25118947,  0.68540622,
       -0.90869318,  1.22861713])
```

In [86]:

```
pd.DataFrame(randn(1000))
```

Out[86]:

	0
0	-0.128523
1	1.371235
2	0.917059
3	0.559066
4	-0.367017
...	...
995	1.158781
996	-0.429806
997	-0.155519
998	0.477970
999	-0.402560

1000 rows × 1 columns

In [87]:

```
pd.date_range('2019-06-07', periods = 1000)
```

Out[87]:

```
DatetimeIndex(['2019-06-07', '2019-06-08', '2019-06-09', '2019-06-10',
               '2019-06-11', '2019-06-12', '2019-06-13', '2019-06-14',
               '2019-06-15', '2019-06-16',
               ...
               '2022-02-21', '2022-02-22', '2022-02-23', '2022-02-24',
               '2022-02-25', '2022-02-26', '2022-02-27', '2022-02-28',
               '2022-03-01', '2022-03-02'],
              dtype='datetime64[ns]', length=1000, freq='D')
```

In [88]:

```
df = pd.DataFrame(randn(1000), index = pd.date_range('2019-06-07', periods = 1000), col
ts = pd.Series(randn(1000), index = pd.date_range('2019-06-07', periods = 1000))
# ts
df.head()
```

Out[88]:

	value
2019-06-07	1.956379
2019-06-08	0.322733
2019-06-09	1.038293
2019-06-10	-0.810037
2019-06-11	-1.023886

In [89]:

```
ts = pd.Series(randn(1000), index = pd.date_range('2019-06-07', periods = 1000))
ts  # this s a series
```

Out[89]:

```
2019-06-07    0.021874
2019-06-08   -2.067731
2019-06-09    1.016978
2019-06-10    1.296927
2019-06-11    0.006605
                ...
2022-02-26   -0.778807
2022-02-27   -0.759849
2022-02-28   -2.121116
2022-03-01    0.197771
2022-03-02    0.607204
Freq: D, Length: 1000, dtype: float64
```

In [90]:

```
df['value'] = df['value'].cumsum()   # cumsum() give the sumation of previous data and
df.head()
```

Out[90]:

	value
2019-06-07	1.956379
2019-06-08	2.279112
2019-06-09	3.317404
2019-06-10	2.507367
2019-06-11	1.483481

In [91]:

```
ts = ts.cumsum()
ts.head()
```

Out[91]:

```
2019-06-07    0.021874
2019-06-08   -2.045857
2019-06-09   -1.028879
2019-06-10    0.268048
2019-06-11    0.274654
Freq: D, dtype: float64
```

In [92]:

```
type(df), type(ts)
```

Out[92]:

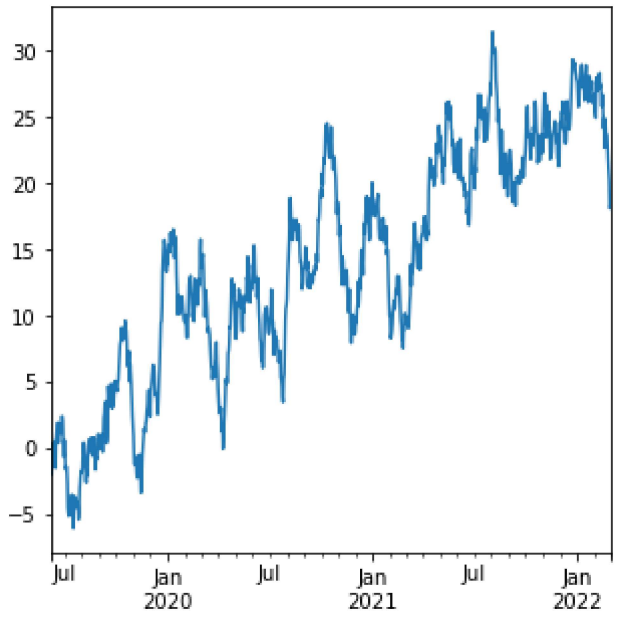
```
(pandas.core.frame.DataFrame, pandas.core.series.Series)
```

In [93]:

```
1 ts.plot(figsize=(5,5))
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c4644070>
```

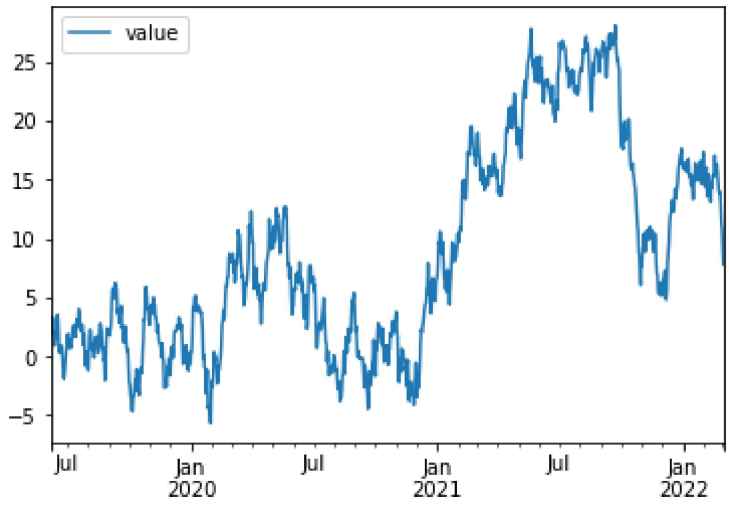


In [94]:

```
1 df.plot()
```

Out[94]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c3488e50>
```

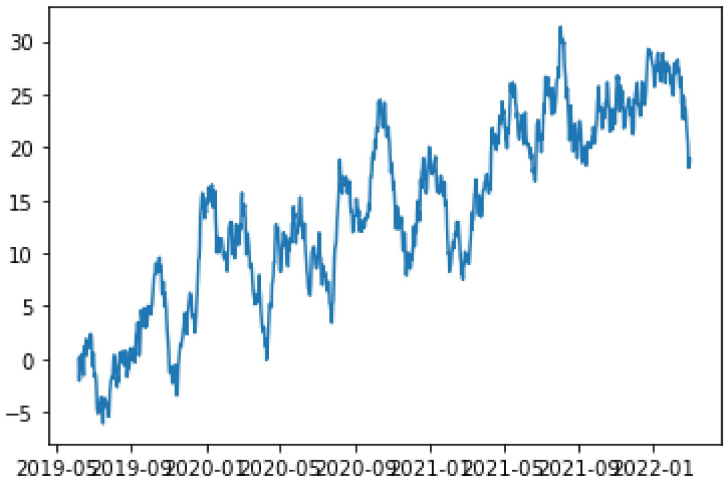


In [95]:

```
1 plt.plot(ts)
```

Out[95]:

```
[<matplotlib.lines.Line2D at 0x2e5c5ac8190>]
```

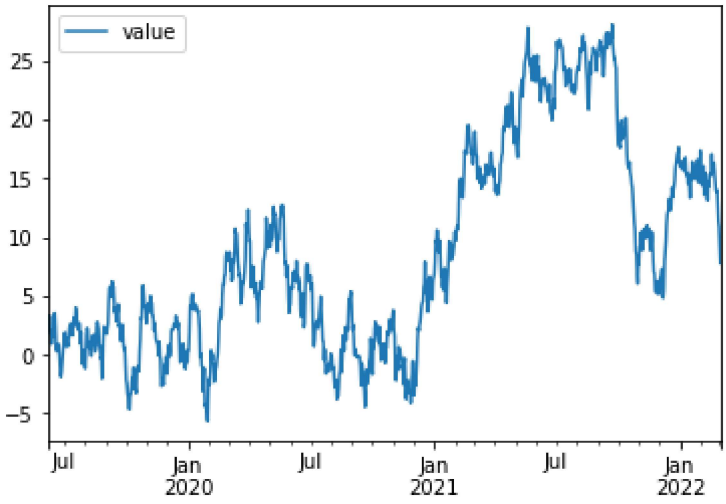


In [96]:

```
1 df.plot()
```

Out[96]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c5b42220>
```



In [97]:

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

Out[97]:

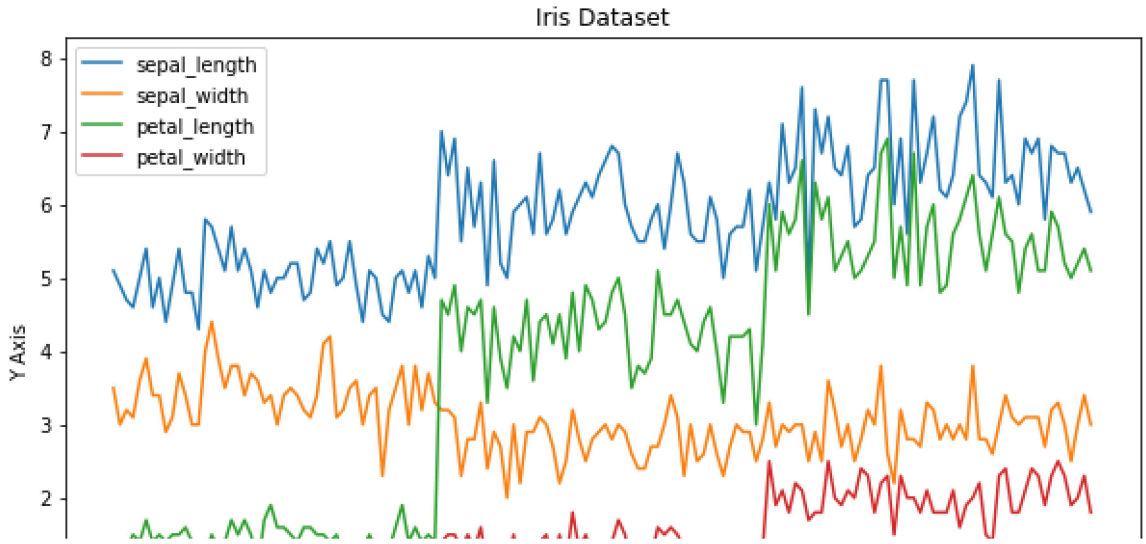
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [98]:

```
ax = iris.plot(figsize=(10,6), title='Iris Dataset')
ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')
```

Out[98]:

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

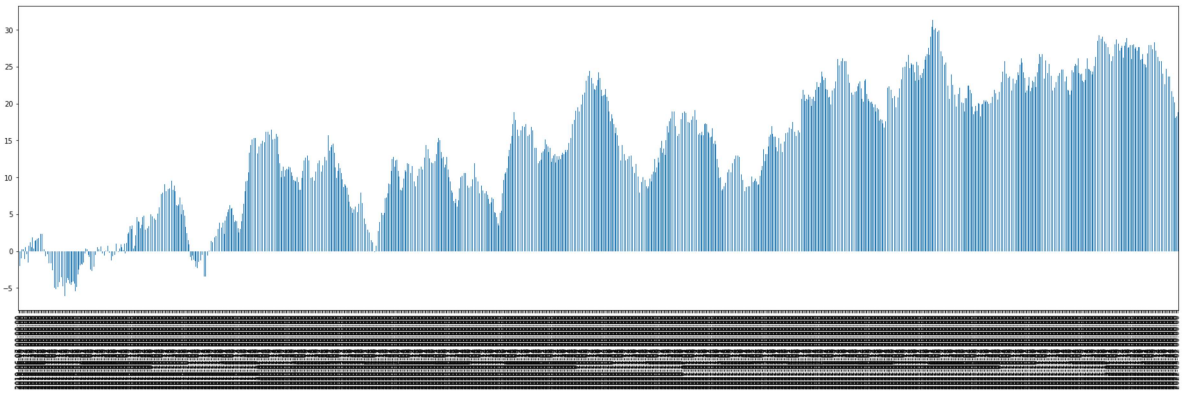


In [99]:

```
1 ts.plot(kind='bar', figsize=(30,8))
```

Out[99]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c5d9a100>
```



In [101]:

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

In [102]:

```
1 df.iloc[0]
```

Out[102]:

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

In [104]:

```
1 df.head()
```

Out[104]:

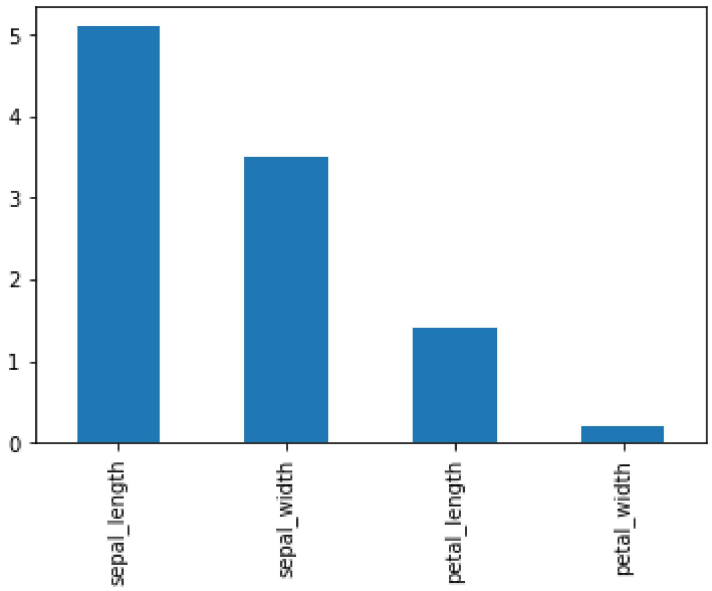
	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [110]:

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

Out[110]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c7c4f730>
```

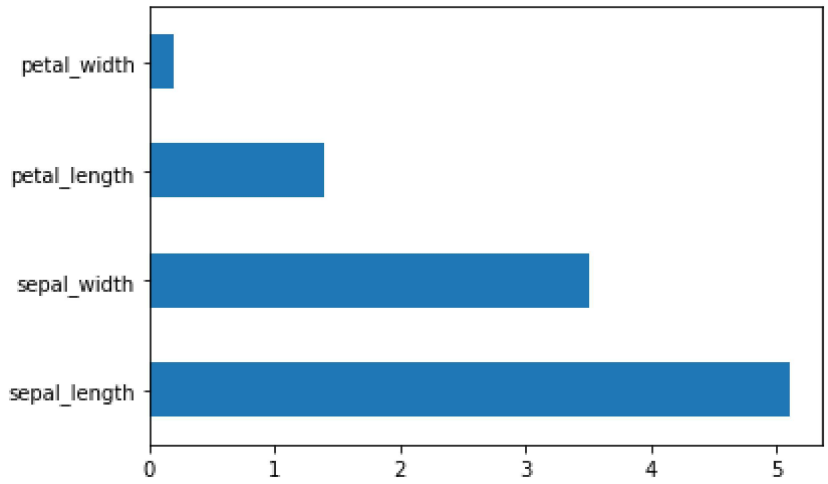


In [112]:

```
df.iloc[0].plot.barh()
```

Out[112]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5c7d03d00>
```



In [113]:

```
titanic = sns.load_dataset('titanic')
```

In [114]:

```
1 titanic.head()
```

Out[114]:

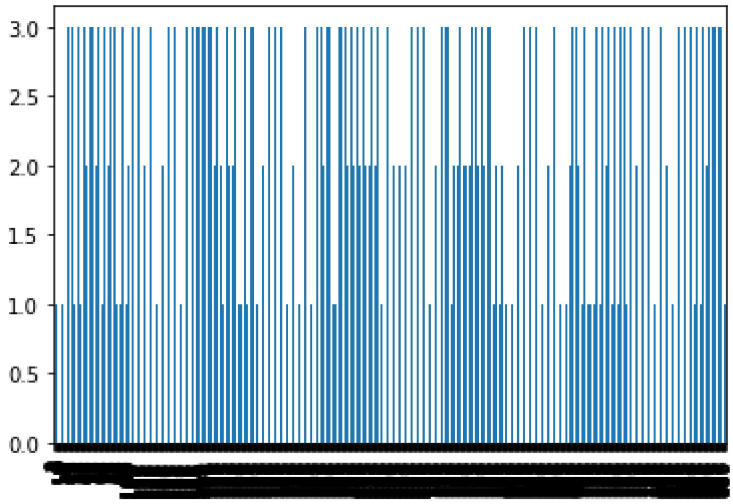
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True

In [117]:

```
1 titanic['pclass'].plot(kind = 'bar')
```

Out[117]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5ccba1d00>
```



In [120]:

```
df = pd.DataFrame(randn(10, 4), columns=['a', 'b', 'c', 'd'])
df.head(10)
```

Out[120]:

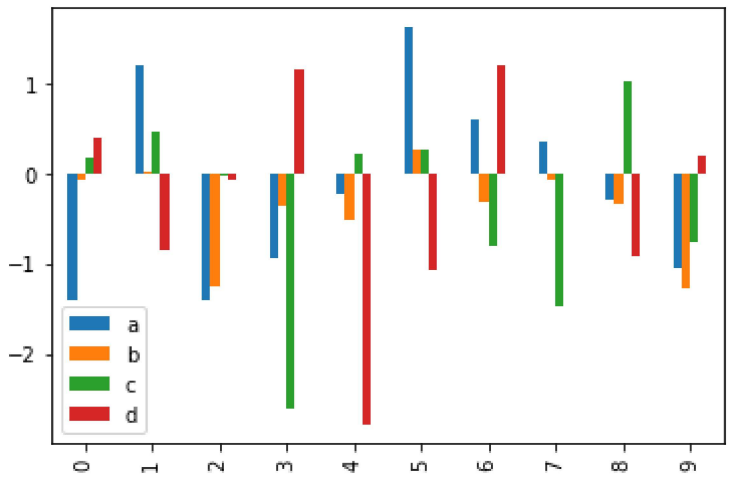
	a	b	c	d
0	-1.403937	-0.065646	0.175011	0.397916
1	1.202219	0.032753	0.474347	-0.845805
2	-1.398518	-1.232025	-0.015288	-0.053669
3	-0.923638	-0.345441	-2.594002	1.166250
4	-0.214687	-0.511333	0.239426	-2.770286
5	1.638403	0.272781	0.280735	-1.067086
6	0.612334	-0.312744	-0.799813	1.204550
7	0.361015	-0.067154	-1.461609	0.010055
8	-0.293154	-0.316666	1.022209	-0.907573
9	-1.051405	-1.265401	-0.758160	0.202970

In [121]:

```
df.plot.bar()
```

Out[121]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5cd9179a0>
```

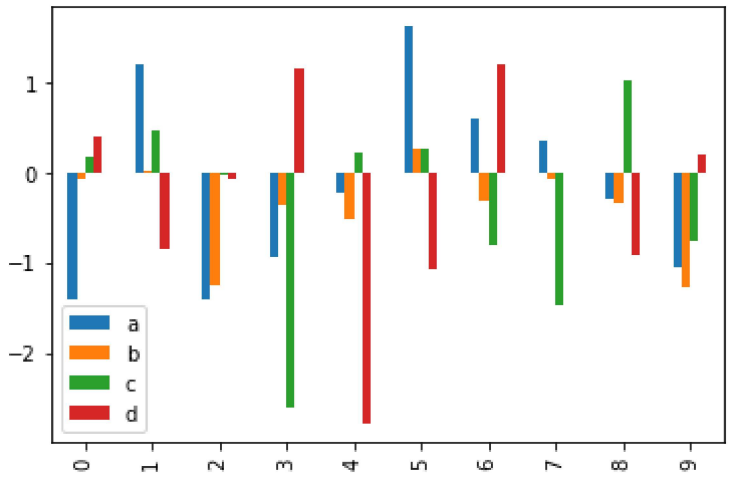


In [122]:

```
1 df.plot(kind = 'bar')
```

Out[122]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5cf9d7ca0>
```

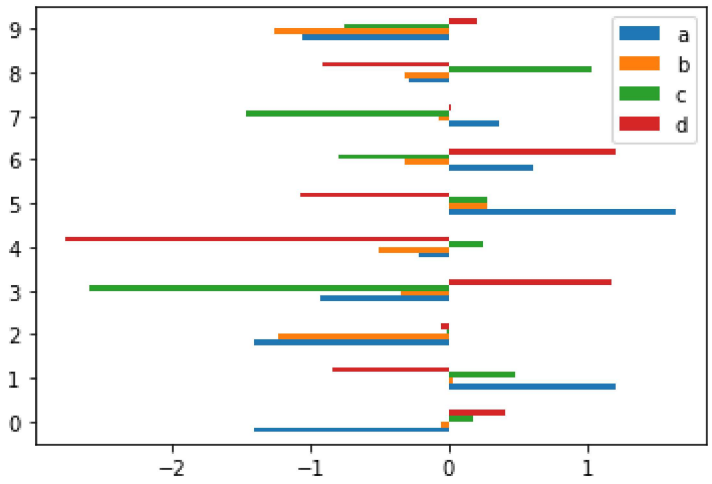


In [123]:

```
1 df.plot.barh()
```

Out[123]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5cfa9e250>
```



In [126]:

```
1 iris.head()
```

Out[126]:

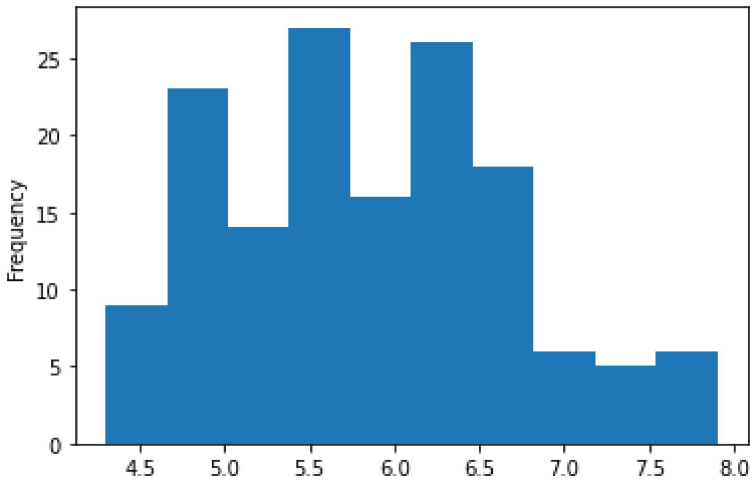
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [129]:

```
1 iris['sepal_length'].plot.hist()
2 #  x take ranges and y give the number of outcomes
```

Out[129]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d0db4d60>
```

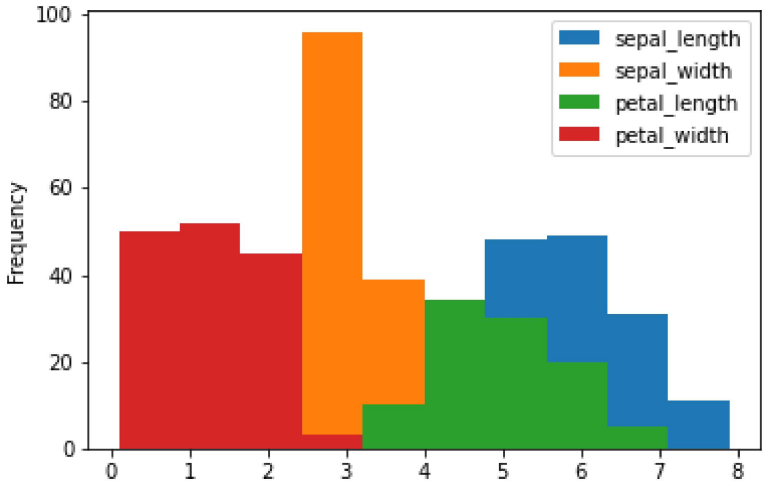


In [130]:

```
1 iris.plot.hist()
```

Out[130]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d0e20040>
```

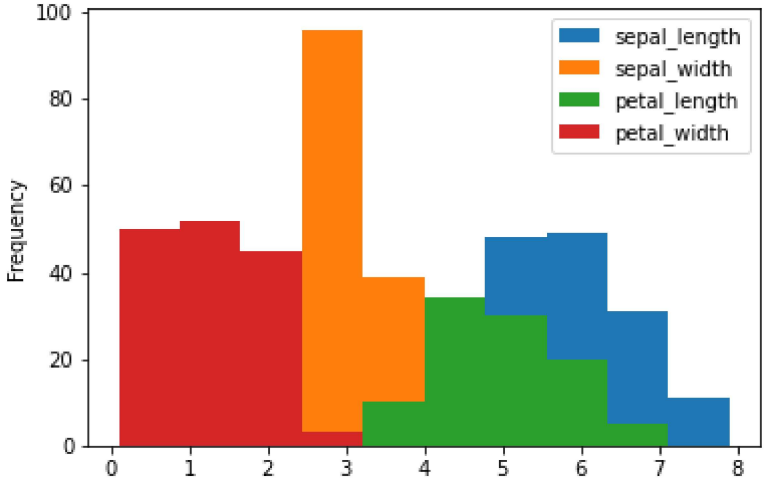


In [125]:

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

Out[125]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5cf9bce50>
```

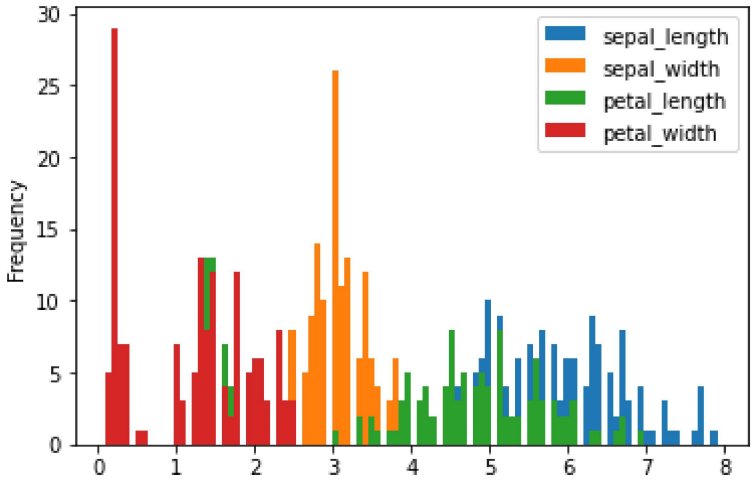


In [135]:

```
iris.plot(kind = 'hist', stacked = False, bins = 100)
# bins controlled the width of graph
# stacked give smaller graph
```

Out[135]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d1caf5e0>
```

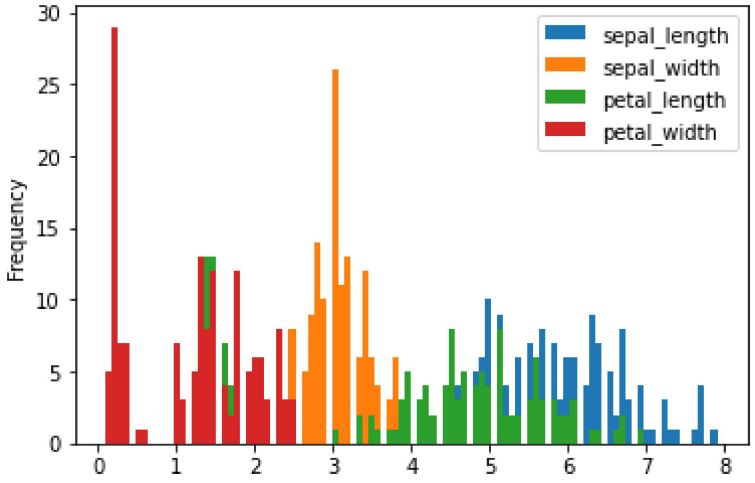


In [139]:

```
iris.plot(kind = 'hist', bins = 100)
```

Out[139]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d28377f0>
```

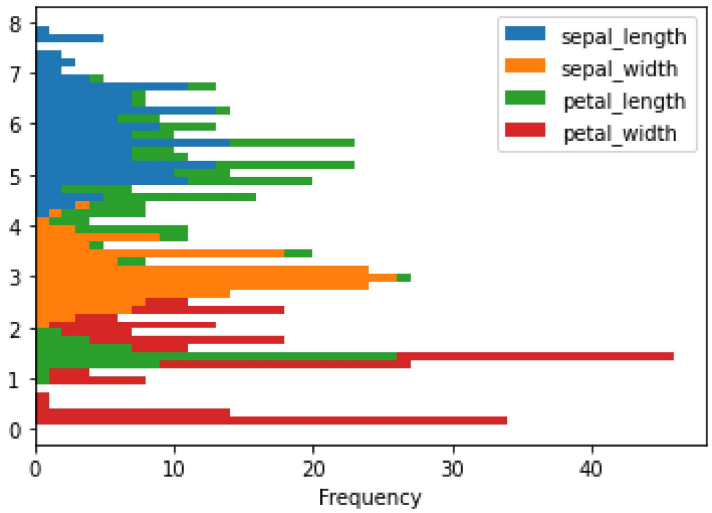


In [140]:

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

Out[140]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d27c7df0>
```



In [153]:

```
1 iris.head()
```

Out[153]:

	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 [142]:

```
iris['sepal_width'].diff()
```

Out[142]:

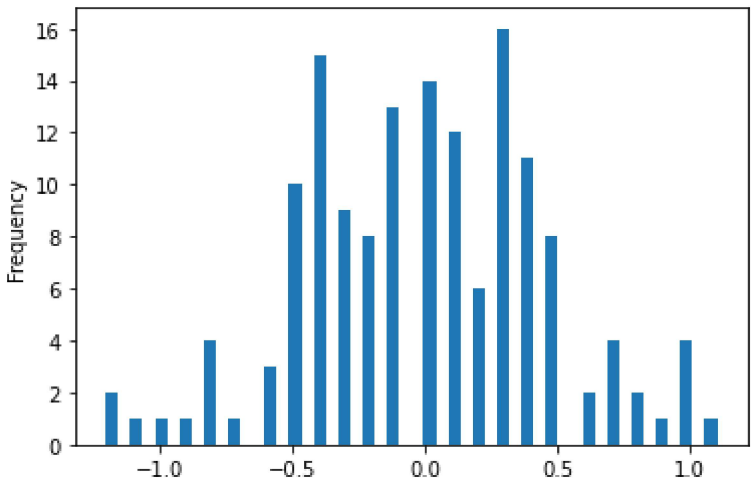
```
0      NaN
1     -0.5
2      0.2
3     -0.1
4      0.5
      ...
145   -0.3
146   -0.5
147    0.5
148    0.4
149   -0.4
Name: sepal_width, Length: 150, dtype: float64
```

In [163]:

```
iris['sepal_width'].diff().plot(kind = 'hist', stacked = True, bins = 50)
# stacked true then smaller graph is in front and large one is in background

```

Out[163]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d57a18e0>
```



In [147]:

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

Out[147]:

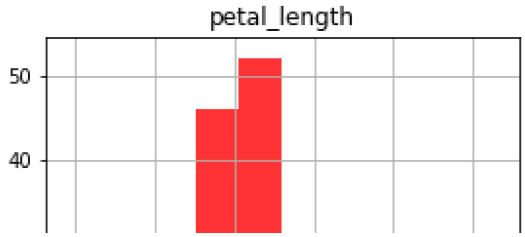
	sepal_length	sepal_width	petal_length	petal_width
0	NaN	NaN	NaN	NaN
1	-0.2	-0.5	0.0	0.0
2	-0.2	0.2	-0.1	0.0
3	-0.1	-0.1	0.2	0.0
4	0.4	0.5	-0.1	0.0

In [151]:

```
1 df.diff().hist(color = 'r', alpha = 0.8, figsize=(10,10))
```

Out[151]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D470EB5
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D48F1CA
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D495C88
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D49850A
0>]],
      dtype=object)
```

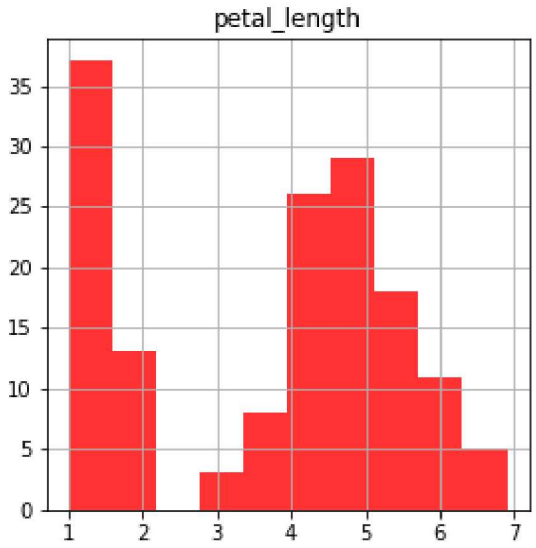


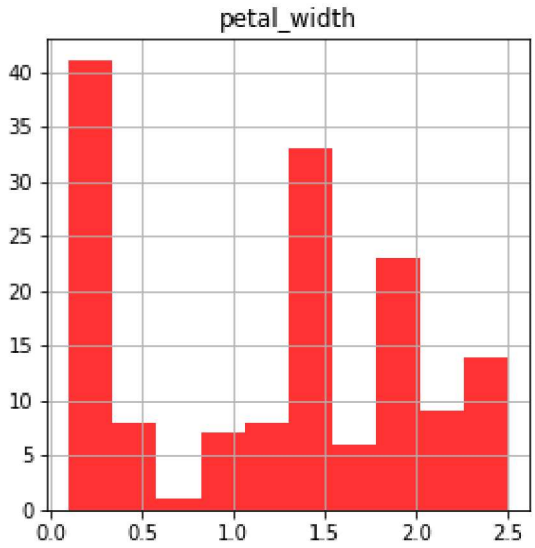
In [155]:

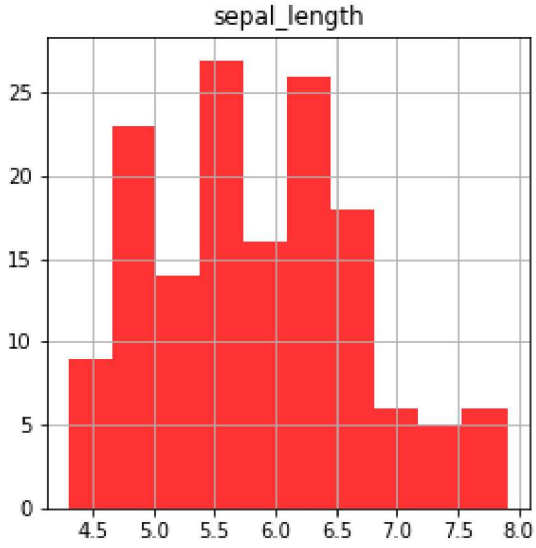
```
iris.hist(color = 'r', alpha = 0.8, figsize=(10,10))
# we can also give hexacode of any colour
```

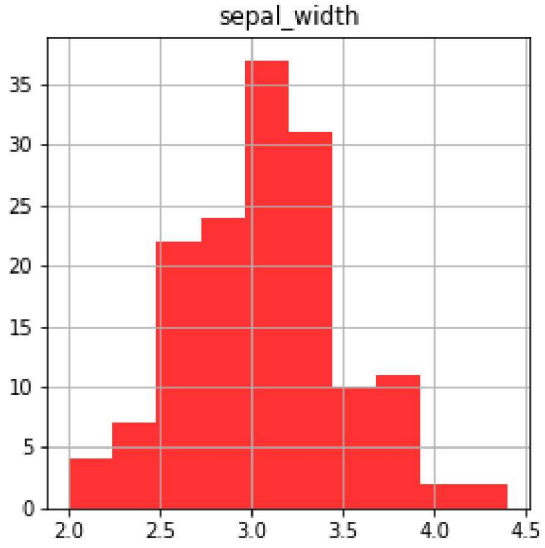
Out[155]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D4F6BB80
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D4FAEBE0
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D500F2B0
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D5038A30
>]],
      dtype=object)
```









In [156]:

```
df
```

Out[156]:

	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 [152]:

```
color = {'boxes': 'DarkGreen', 'whiskers': 'b'}
color
```

Out[152]:

```
{'boxes': 'DarkGreen', 'whiskers': 'b'}
```

In [157]:

```
1 df
```

Out[157]:

	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

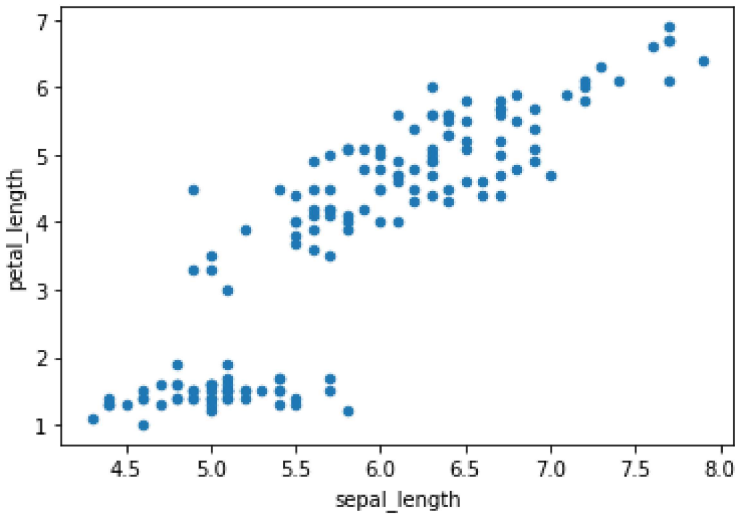
150 rows × 4 columns

In [159]:

```
1 df.plot.scatter(x = 'sepal_length', y = 'petal_length')
2 #  take 2 column and compare
```

Out[159]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d561c040>
```

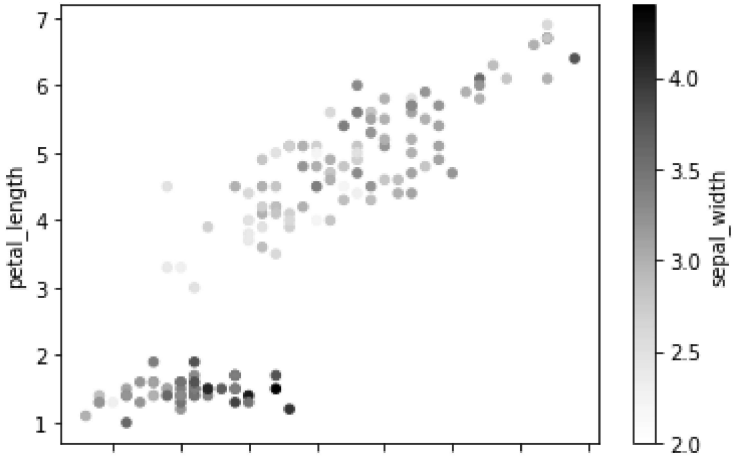


In [164]:

```
df.plot.scatter(x = 'sepal_length', y = 'petal_length', c = 'sepal_width')
# third one is representing colour
# c is taken as colour parameter
```

Out[164]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d5875100>
```



In [165]:

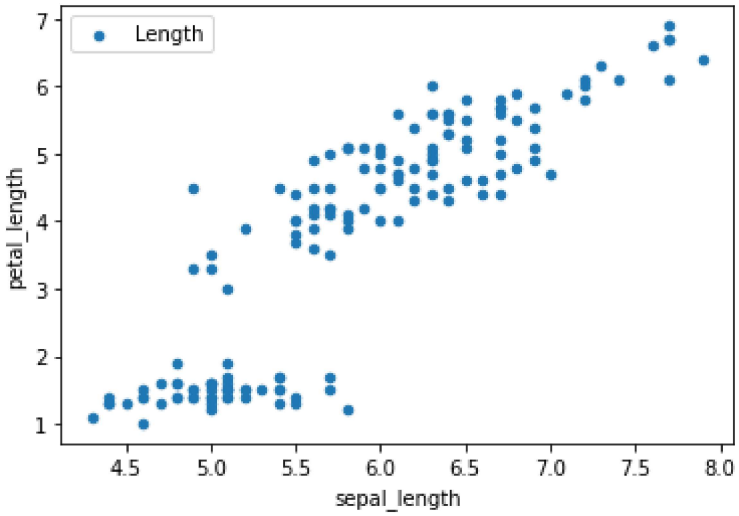
```
df.head()
```

Out[165]:

	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

In [166]:

```
df.plot.scatter(x = 'sepal_length', y = 'petal_length', label = 'Length');
#df.plot.scatter(x = 'sepal_width', y = 'petal_width', label = 'Width', ax = ax, color
#df.plot.scatter(x = 'sepal_width', y = 'petal_length', label = 'Width', ax = ax, color
```



In [ ]:

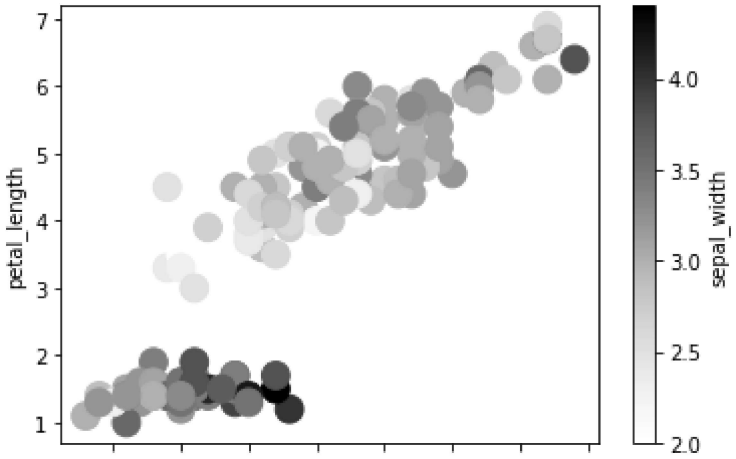
```
```

In [171]:

```
df.plot.scatter(x = 'sepal_length', y = 'petal_length', c = 'sepal_width', s = 190)
# s set the size of dots
```

Out[171]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d6b830d0>
```



In [ ]:

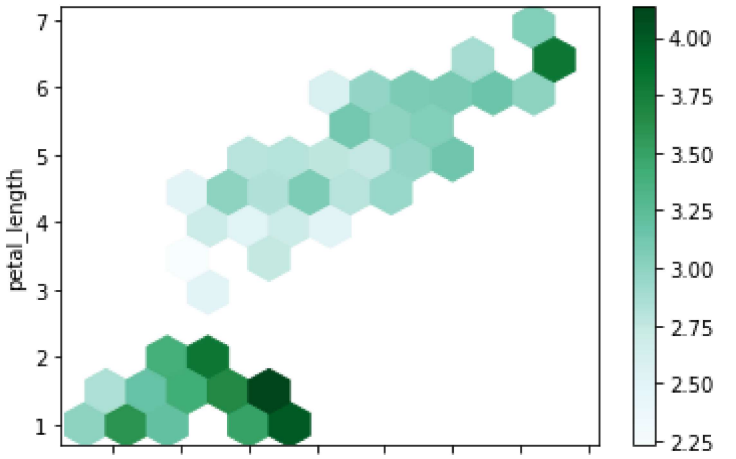
```

```

In [178]:

```
df.plot.hexbin(x = 'sepal_length', y = 'petal_length', gridsize = 12, C = 'sepal_width'
# gridsize set the size of grid
```

Out[178]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d6f929d0>
```



In [ ]:

```

```

In [180]:

```
1 iris.head()
```

Out[180]:

	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 [181]:

```
1 d = df.iloc[0]
2 d
```

Out[181]:

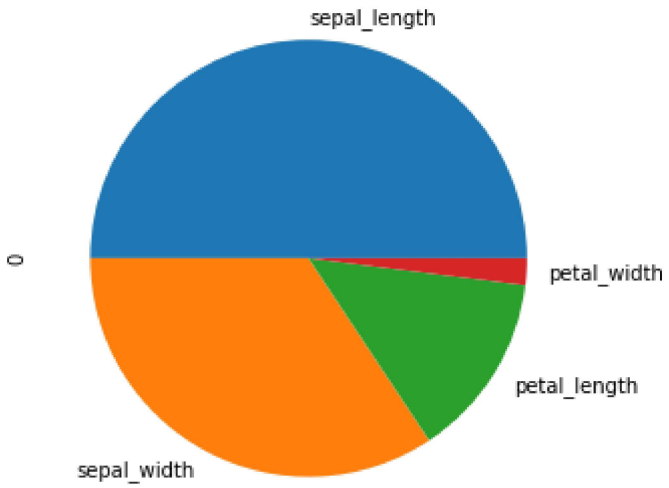
```
sepal_length    5.1
sepal_width     3.5
petal_length    1.4
petal_width     0.2
Name: 0, dtype: float64
```

In [183]:

```
1 d.plot.pie(figsize = (5,5))
```

Out[183]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d7071bb0>
```



In [185]:

```
1 df.head()
```

Out[185]:

	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

In [191]:

```
1 d = df.head(4).T
```

In [192]:

```
1 d
```

Out[192]:

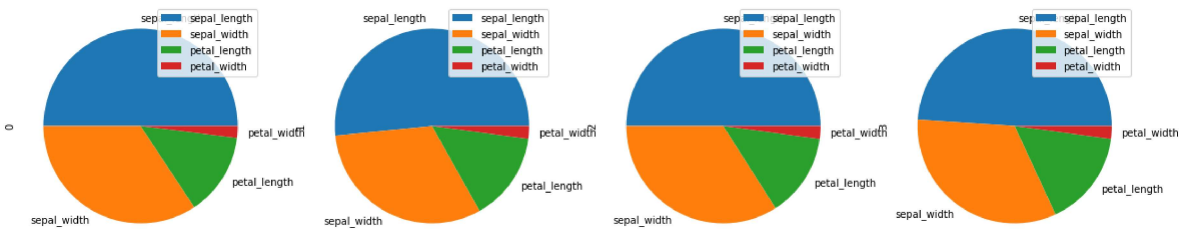
	0	1	2	3
sepal_length	5.1	4.9	4.7	4.6
sepal_width	3.5	3.0	3.2	3.1
petal_length	1.4	1.4	1.3	1.5
petal_width	0.2	0.2	0.2	0.2

In [196]:

```
1 d.plot.pie(subplots = True, figsize = (20, 20))
```

Out[196]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D75AAAC0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D75D0CA0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D7733310>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D775D8B0
>],
      dtype=object)
```

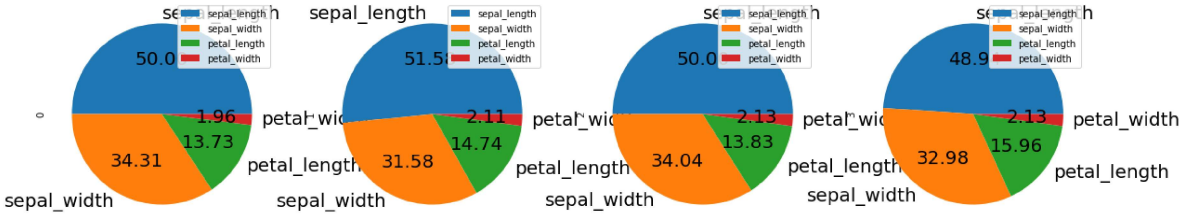


In [200]:

```
d.plot.pie(subplots = True, figsize = (20, 20), fontsize = 20, autopct = '%.2f')
# also change the font size
#  autopct is a part of beautification
```

Out[200]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D7F4DFD0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D80C82B0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D80E94F0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5D8118610
>],
      dtype=object)
```



In [202]:

```
[0.1]*4
```

Out[202]:

```
[0.1, 0.1, 0.1, 0.1]
```

In [204]:

```
series = pd.Series([0.2]*5, index = ['a','b','c', 'd','e'], name = 'Pie Plot')
series
# series.plot.pie()
```

Out[204]:

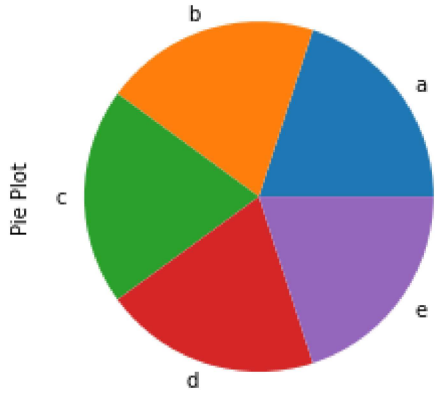
```
a    0.2
b    0.2
c    0.2
d    0.2
e    0.2
Name: Pie Plot, dtype: float64
```

In [205]:

```
1 series.plot.pie()
```

Out[205]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5d8391370>
```



In [ ]:

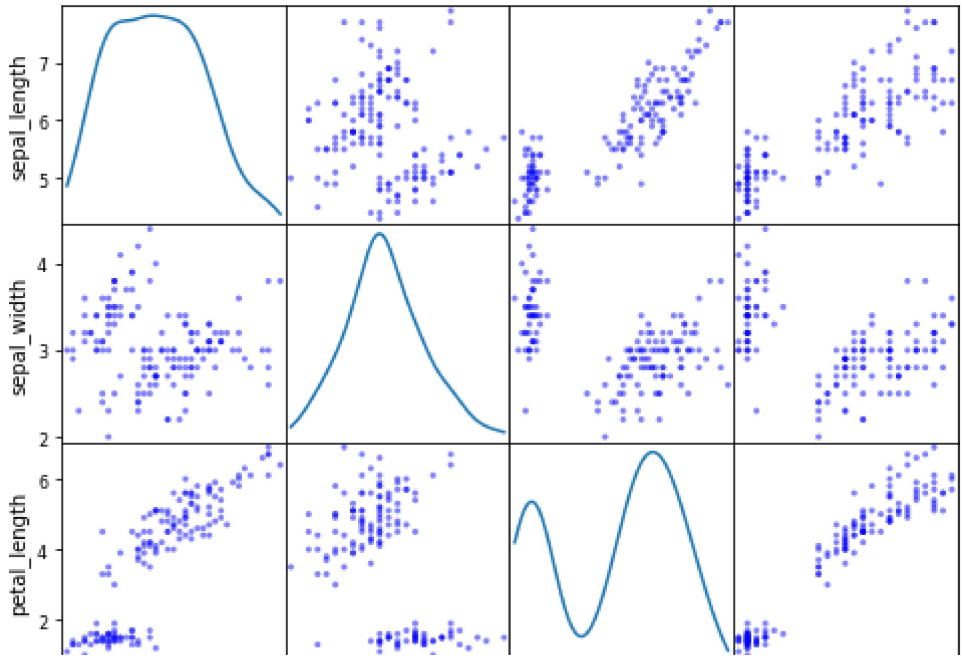
```
1
```

In [206]:

```
1 from pandas.plotting import scatter_matrix
```

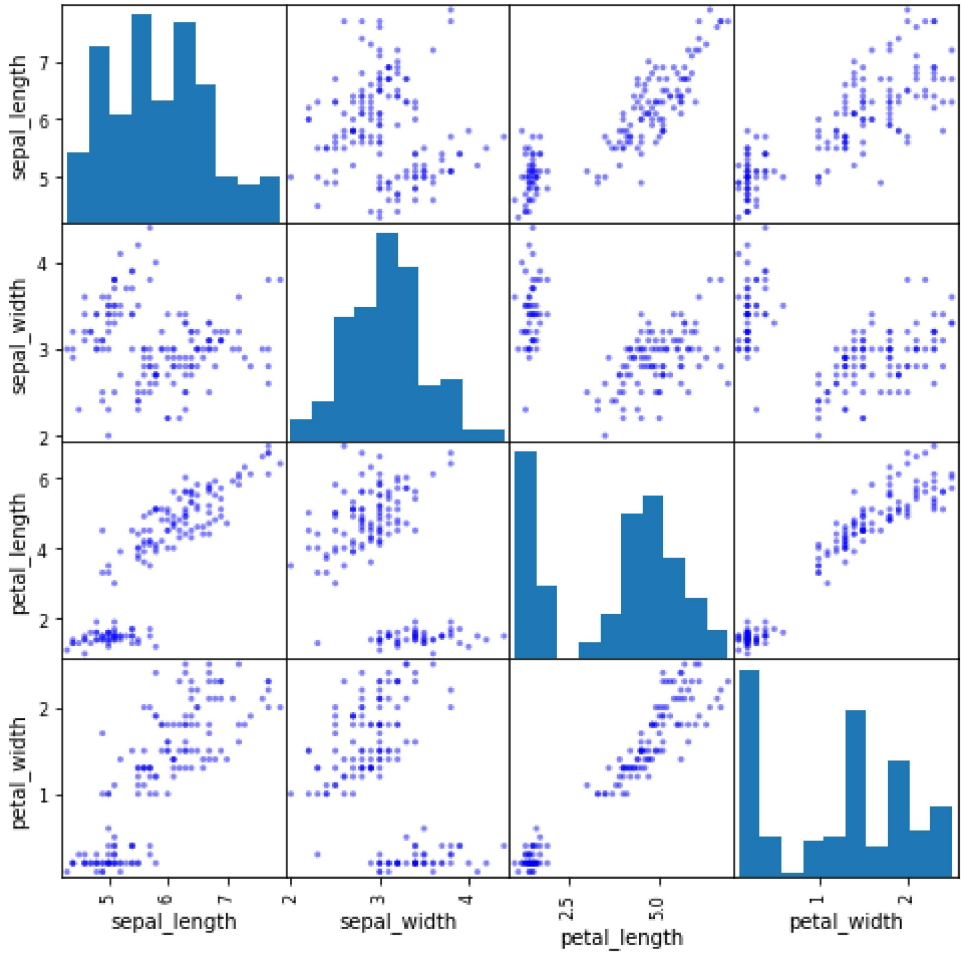
In [212]:

```
scatter_matrix(df, figsize= (8,8), diagonal='kde', color = 'b')
plt.show()
# kde is kernal desity estimation  --> normal distribution plot
```



In [210]:

```
scatter_matrix(df, figsize= (8,8), color = 'b')
plt.show()
```

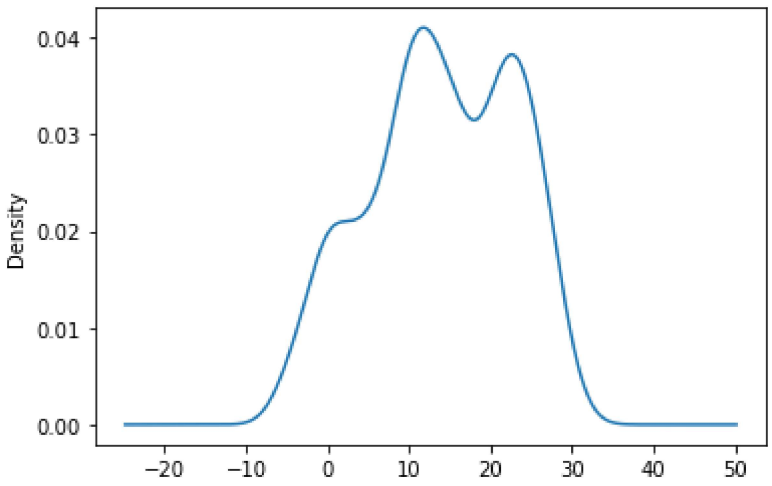


In [213]:

```
1 ts.plot.kde()
```

Out[213]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dbe17d90>
```



In [ ]:

```
1
```

In [215]:

```
1 from pandas.plotting import andrews_curves
```

In [216]:

```
1 andrews_curves(df, 'sepal_width')
```

Out[216]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dbcd5a30>
```

In [ ]:

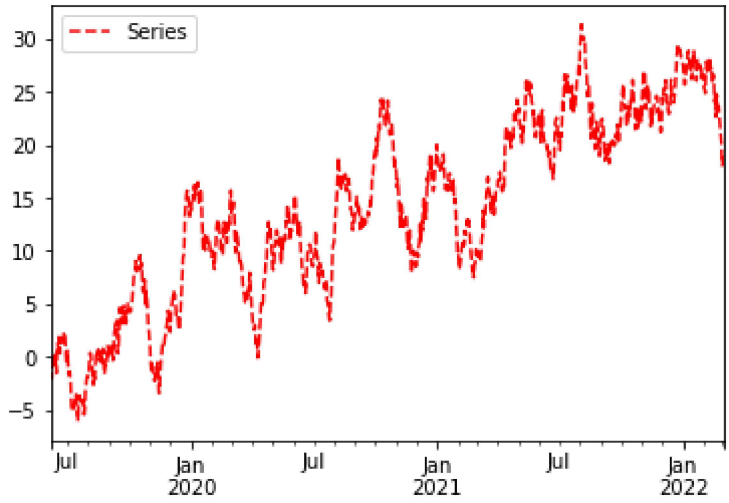
```
1
```

In [218]:

```
1 ts.plot(style = 'r--', label = 'Series', legend = True)
```

Out[218]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dbcc3dc0>
```

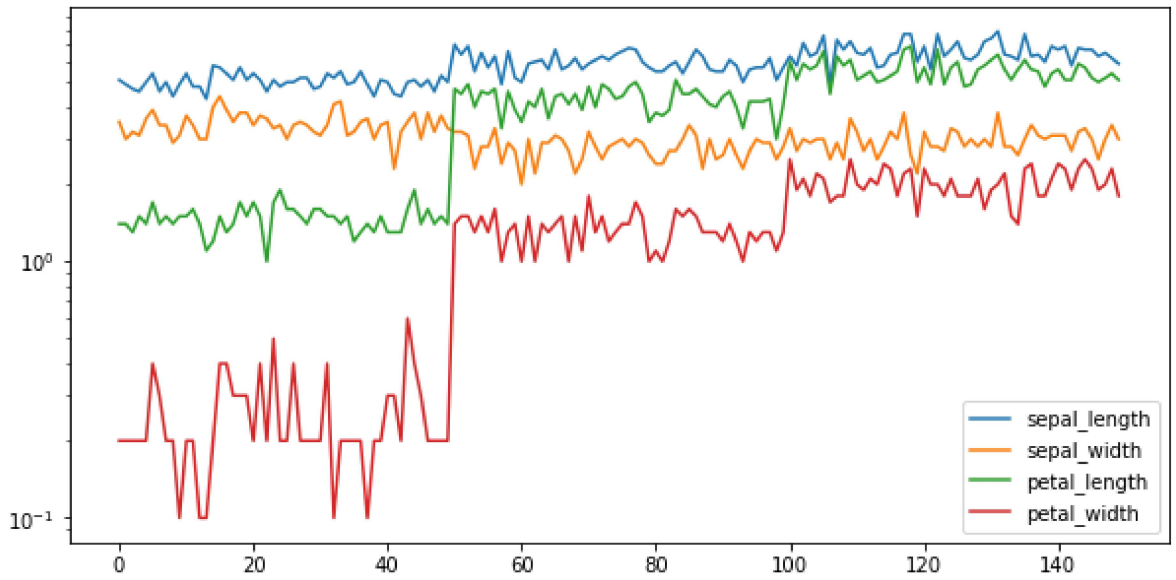


In [221]:

```
1 df.plot(legend = True, figsize = (10, 5), logy = True)
2 # logy take logarithimic and then plot
```

Out[221]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dc179f40>
```



In [223]:

```
df.head(0)
```

Out[223]:

sepal_length	sepal_width	petal_length	petal_width

In [ ]:

```

```

In [224]:

```
x = df.drop(['sepal_width', 'petal_width'], axis = 1)
x.head()
```

Out[224]:

	sepal_length	petal_length
0	5.1	1.4
1	4.9	1.4
2	4.7	1.3
3	4.6	1.5
4	5.0	1.4

In [225]:

```
y = df.drop(['sepal_length', 'petal_length'], axis = 1)
y.head()
```

Out[225]:

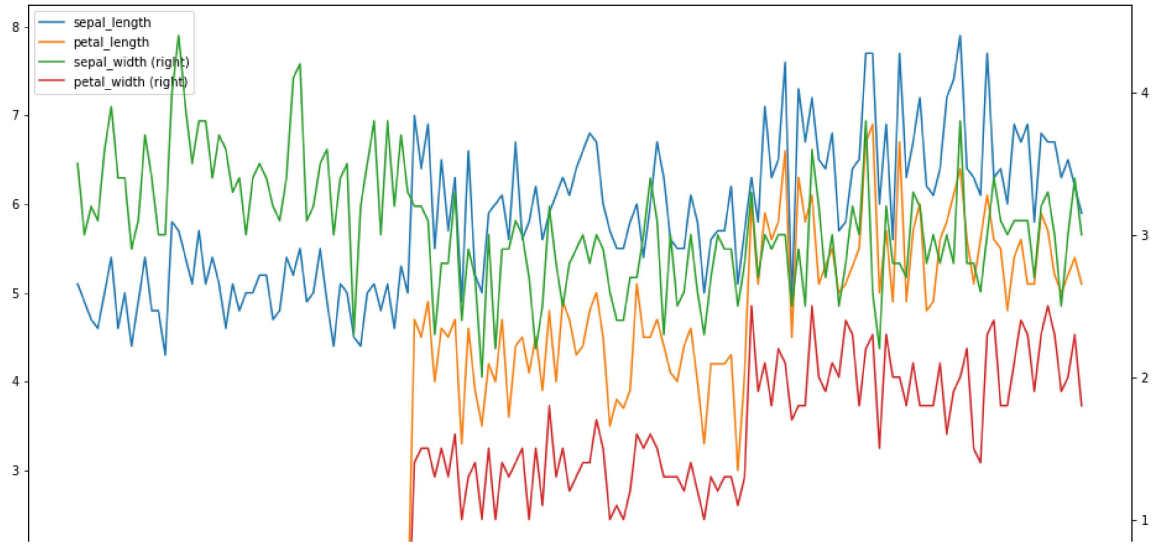
	sepal_width	petal_width
0	3.5	0.2
1	3.0	0.2
2	3.2	0.2
3	3.1	0.2
4	3.6	0.2

In [227]:

```
ax = x.plot()
y.plot(figsize = (16,10), secondary_y=True, ax = ax)
```

Out[227]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dd281fa0>
```



In [ ]:

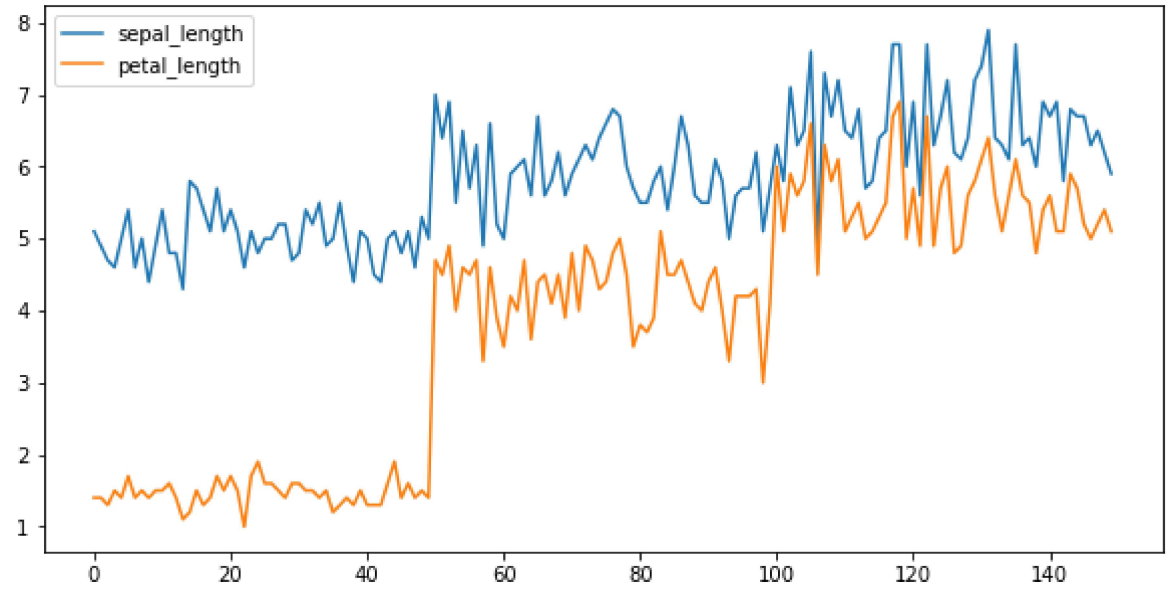
```

```

In [228]:

```
x.plot(figsize=(10,5), x_compat = True)
```

Out[228]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e5dd6bce80>
```



In [ ]:

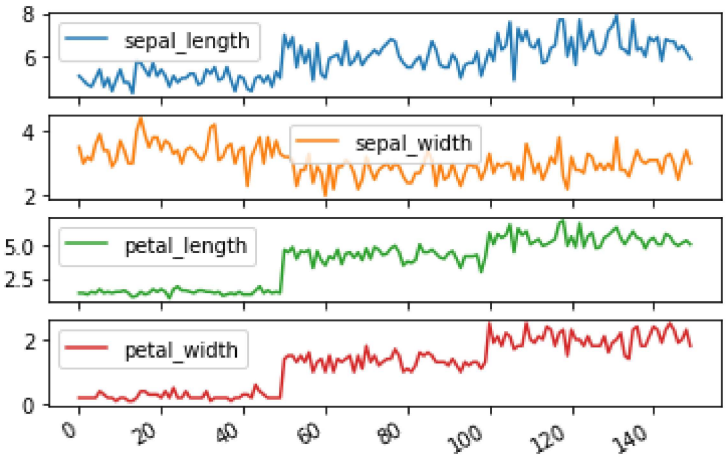
```

```

In [229]:

```
df.plot(subplots = True)

```

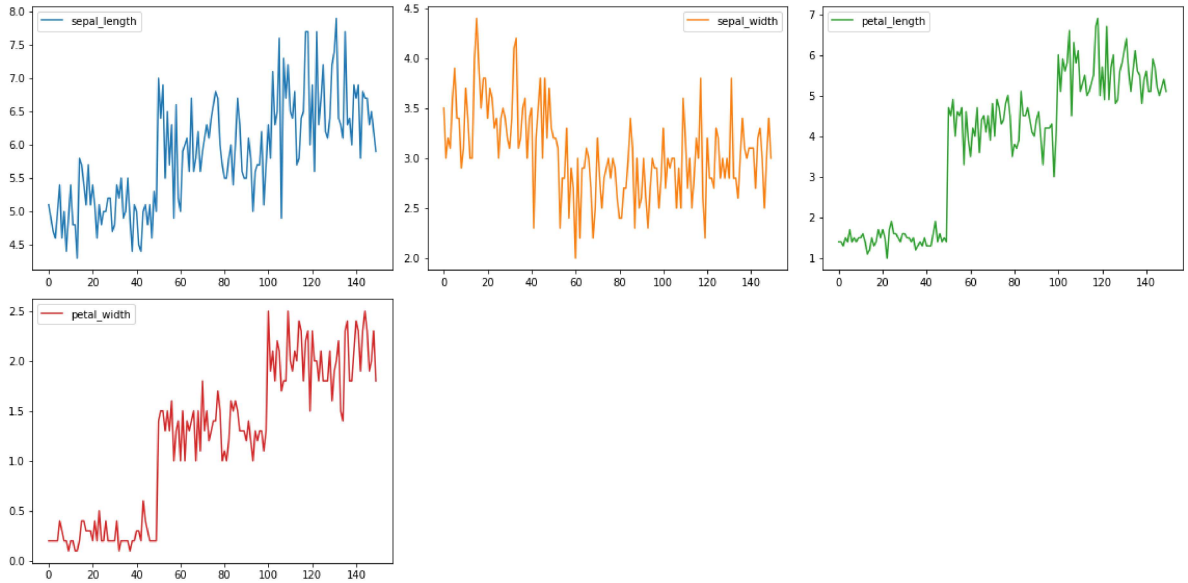
Out[229]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002E5DD640820>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5DD65FFD0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5DD909640>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002E5DD933B80
>],
      dtype=object)
```



In [230]:

```
df.plot(subplots = True, sharex = False, layout = (2,3), figsize = (16,8))
plt.tight_layout()
```



In [ ]:

```

```

In [239]:

```
1 from descartes import PolygonPatch
```

In [ ]:

```
1
```

In [237]:

```
1 map_df = gpd.read_file("London_Borough_Excluding_MHW.shp")
2 # check data type so we can see that this is not a normal dataframe, but a GEOdataframe
3 map_df.head()
4 map_df.plot()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-237-335b3608a8ad> in <module>
----> 1 map_df = gpd.read_file("London_Borough_Excluding_MHW.shp")
      2 # check data type so we can see that this is not a normal dataframe,
but a GEOdataframe
      3 map_df.head()
      4 map_df.plot()

AttributeError: 'function' object has no attribute 'read_file'
```

In [238]:

```
1 import geopandas as gpd
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-238-a62d01c1d62e> in <module>
----> 1 import geopandas as gpd

ModuleNotFoundError: No module named 'geopandas'
```

In [80]:

```
1 map_df
```

Out[80]:

	NAME	GSS_CODE	HECTARES	NONLD_AREA	ONS_INNER	SUB_2009	SUB_2006	geometry
0	Kingston upon Thames	E09000021	3726.117	0.000	F	None	None	POLYGON ((516401.6 160201.8, 516407.3 160210.5...
1	Croydon	E09000008	8649.441	0.000	F	None	None	POLYGON ((535009.2 159504.7, 535005.5 159502, ...
2	Bromley	E09000006	15013.487	0.000	F	None	None	POLYGON ((540373.6 157530.4, 540361.2 157551.9...
								POLYGON ((521975.8

In [ ]:

```
1
```