Technique Assignment 1: Python

Cogs 109

Xing Hong

PID: A15867895

In [1]:
```
# load matplotlib for plotting
# import datasets so we can use the Fisher's iris dataset
import matplotlib.pyplot as plt
import matplotlib
from sklearn import datasets
```

1. Loops in Python

Result:

In [2]:
```
s = 0
x = 3
while x<100:
    s += x
    x += 3
print(s)
```

```
1683
```

2. List operations

Result:

In [3]:
```
s = 0
for i in list(range(3,100,3)):
    s+=i
print(s)
```

```
1683
```

3. Strings

In [4]:
```
string = "Hello Python"
sub = string[:5]
sub2 = string[6:]
sub3,sub4 = string.split()
```

```
print(sub,sub2)
print(sub3,sub4)
```

```
Hello Python
Hello Python
```

4. Datasets: Fisher's iris data

In [5]:
```
# load Fisher's iris dataset
iris = datasets.load_iris()
```

In [6]:
```
# check out the data (we can look at small datasets this way, but
not big ones)
#iris.data

# Once you've looked at the data, be sure to comment out the code
above.
# You should not display unneccessary data in your pdf report,
# especially if it's a long list of numbers!
```

In [7]:
```
# This is part of the dataset too.
#iris.target

# Don't display this in your report either!
```

In [8]:
```
#print(iris.DESCR)
```

5. Plotting data

a. Three classes of iris,50 of each with 4 numeric, predictive attributes and classes: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm class:Iris-Setosa, Iris-Versicolour, Iris-Virginica

b. 4 variables: sepal length, sepal width, petal length, petal width

In [9]:
```
iris.feature_names
```

Out[9]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

c. 150 samples

In [10]:
```
len(iris.target)
```

Out[10]:
```
150
```

d. Yes, three catagories: 'setosa', 'versicolor', 'virginica'and labled samples with 0,1,2

In [11]:
```
iris.target_names
```

Out[11]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [13]:
```
#iris.target
```

e. Not really, they are just raw numbers that looks similar to each other

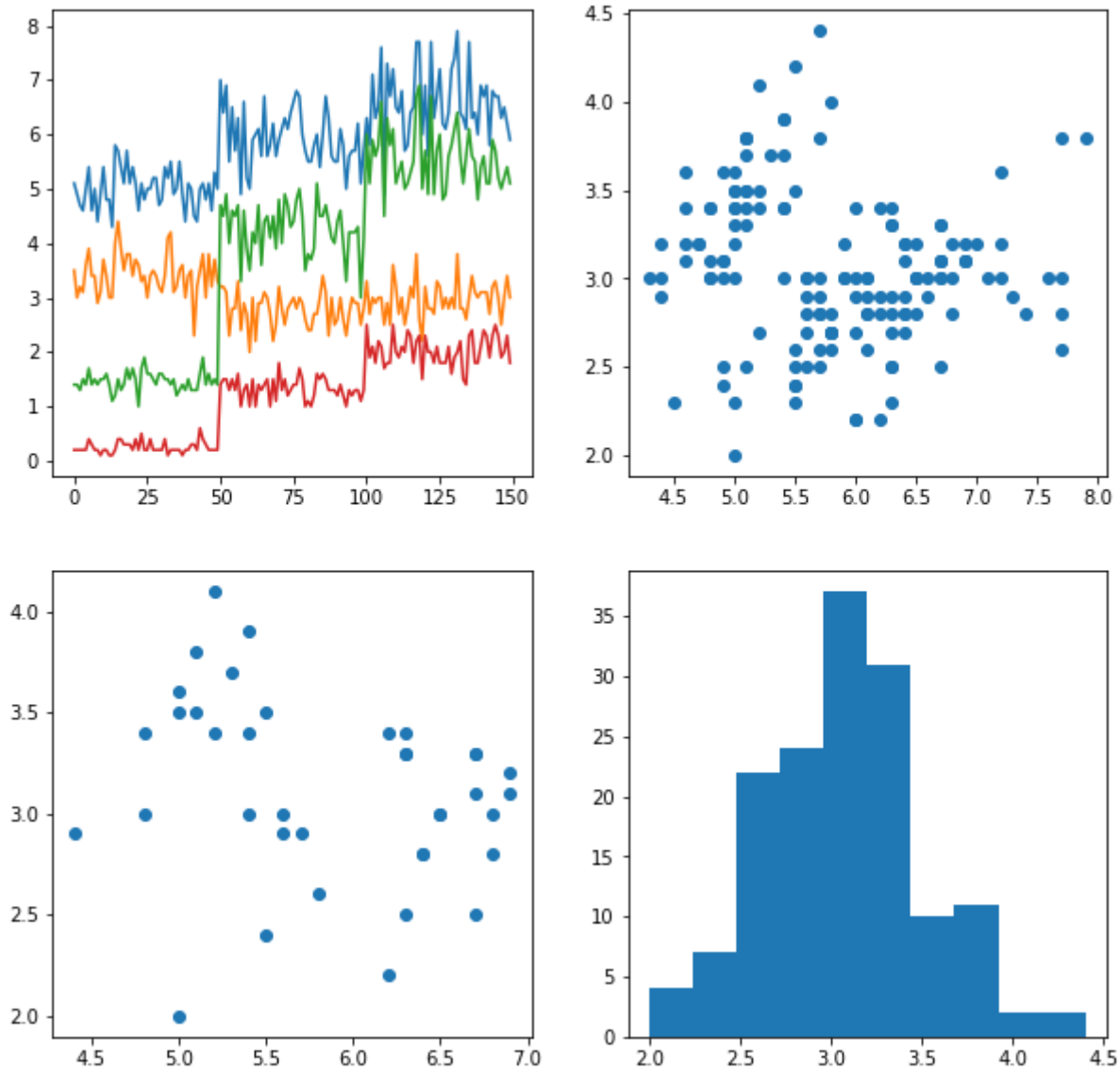
In [14]:
```
plt.rcParams['figure.figsize'] = [10,10]
fig, axs = plt.subplots(2,2)
axs[0,0].plot(iris.data)

x = iris.data[:,0]
y = iris.data[:,1]
axs[0,1].scatter(x,y)

x1 = x[0::4]
y1 = y[0::4]
axs[1,0].scatter(x1,y1)

axs[1,1].hist(y)
```

Out[14]:
```
(array([ 4.,  7., 22., 24., 37., 31., 10., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```



a. x: different samples, 150 in total; differennt lines respresents different features, which are sepal length, sepal width, petal length, petal width.

b. Since there are 50 samples for each kinnd iris,there are dramatic rises in features at the 50,100 of x axis.

c.above

d.There seems to be two cluster/lines in the plot, which may implies the connection between first and second feature.

e.above

f.It has the shape of normal distribution, but slightly slip left, which implies more iris has size around 3.0-3.5, and very less smaller than 2.5 and bigger than 4.0.

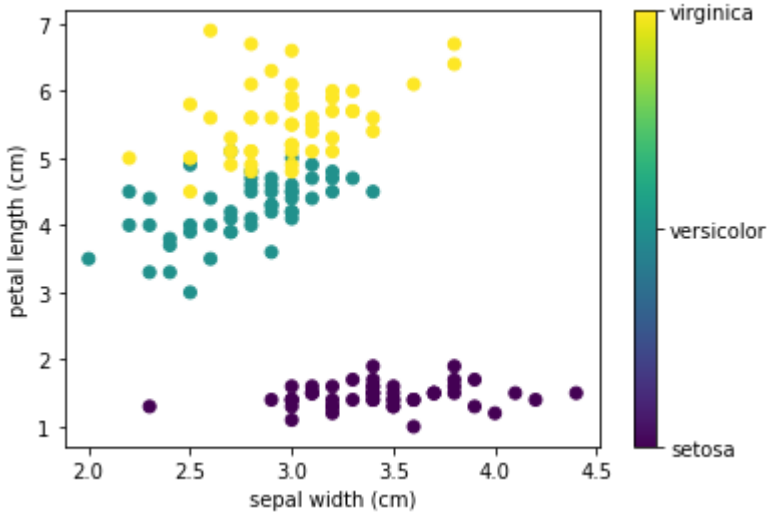
6. Plotting with labels

In [15]:
```
x = 1
y = 2

formatter = plt.FuncFormatter(lambda i, args:
```

```
iris.target_names[i])
plt.figure(figsize=(6, 4))
plt.scatter(iris.data[:, x], iris.data[:, y], c = iris.target)
plt.xlabel(iris.feature_names[x])
plt.ylabel(iris.feature_names[y])

plt.colorbar(ticks=[0, 1, 2], format=formatter)
plt.show()
```



In [ ]: