Technique assignment 1

Cogs 109 Spring 2020

Er Lin
A16140839

In [1]:
```
# load matplotlib for plotting
# import datasets so we can use the Fisher's iris dataset

import matplotlib.pyplot as plt
from sklearn import datasets

import numpy as np
```

1. Calculate 3 + 6 + ... + 99

Type result here:

In [2]:
```
total = 0 # total to keep track of the sum
x = 3 # starting number

# loop up to 100
while x < 100:
    total += x # add x to the total sum
    x += 3 # increase x by 3

total
```

Out[2]:
```
1683
```

2. Calculate 3 + 6 + ... + 99 using range()

Type result here:

In [3]:
```
# create list with range from 3 to 100 with step 3 and perform sum
on list
sum(range(3, 100, 3))
```

Out[3]:
```
1683
```

3. Strings

In [4]:
```
# create string
s = 'Hello everyone in COGS109'
s
```
Out[4]:
```
'Hello everyone in COGS109'
```

In [5]:
```
# get substring hello
hello = s[:s.index('o') + 1]
hello
```
Out[5]:
```
'Hello'
```

In [6]:
```
# get substring everyone
everyone = s[s.index('everyone'):s.index('everyone') + len('everyon
e')]
everyone
```
Out[6]:
```
'everyone'
```

In [7]:
```
# get substring COG109
COGS109 = s[s.index('C'):]
COGS109
```
Out[7]:
```
'COGS109'
```

4. Datasets: Fisher's iris data

In [8]:
```
# load Fisher's iris dataset
iris = datasets.load_iris()
```

In [9]:
```
# check out the data (we can look at small datasets this way, but n
ot big ones)
# iris.data
# Once you've looked at the data, be sure to comment out the code a
bove.
# You should not display unneccessary data in your pdf report,
# especially if it's a long list of numbers!
```

In [10]:
```
# This is part of the dataset too.
# iris.target
# Don't display this in your report either!
```

```
In [11]: len(iris.target) == len(iris.data)

Out[11]: True
```

```
In [12]: iris.feature_names

Out[12]: ['sepal length (cm)',
          'sepal width (cm)',
          'petal length (cm)',
          'petal width (cm)']
```

```
In [13]: iris.target_names

Out[13]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

a. List and describe the data that has been loaded (hint: there are two arrays).

- The data loaded is a collection of 4 different measurements of 3 different species of the Iris flower:
  - The measurements of each flower is given by the iris.data (array of arrays of 4 elements) where each array contains the following information: ['sepal length','sepal width','petal length','petal width'].
  - We know that each array in iris.data corresponds to an Iris flower species given by iris.target which is an array of the same size as iris.data containing the numbers 0, 1, 2, each representing a different species of Iris flower: ['setosa', 'versicolor', 'virginica'].

b. How many variables are being measured in this data set?

- There are 4 variables ['sepal length','sepal width','petal length','petal width'] for each of the 3 species of Iris flower ['setosa', 'versicolor', 'virginica'].

c. How many samples are there of each measurement?

- There are 150 samples of each measurement.

d. Is there a label for each sample?

- Yes there is a label for each sample given by iris.target which corresponds to the species of Iris flower that the sample belongs to.

e. Looking at the numbers in the iris.data array, can you spot any trends in the data? If so, describe any trends (patterns) you notice.

- The numbers tend to increase when we change to another Iris flower species (moving towards higher indexes in the array).
- The measurements of flowers with label 0 are generally smaller than label 1 and label 1 flowers are smaller than label 2. There are exceptions, but this is the general trend, that I can observe.

5. Plotting data

a. The x-axis represents each flower in iris.data (150 flowers) and the y-axis is the corresponding value of each measurement.
That is why we get 4 lines, one for each of the 4 different measurements per flower:

- Blue line: sepal length
- Orange line: sepal width
- Green line: petal length
- Red line: petal width

b. For sepal length (blue), petal length (green) and petal width (red) the measurements tend to be larger as we move across the different species of iris flowers, specifically:

- setosa < versicolor < virginica (refering to the measurments).

The sepal width stays similar across the 3 different species of iris flowers.

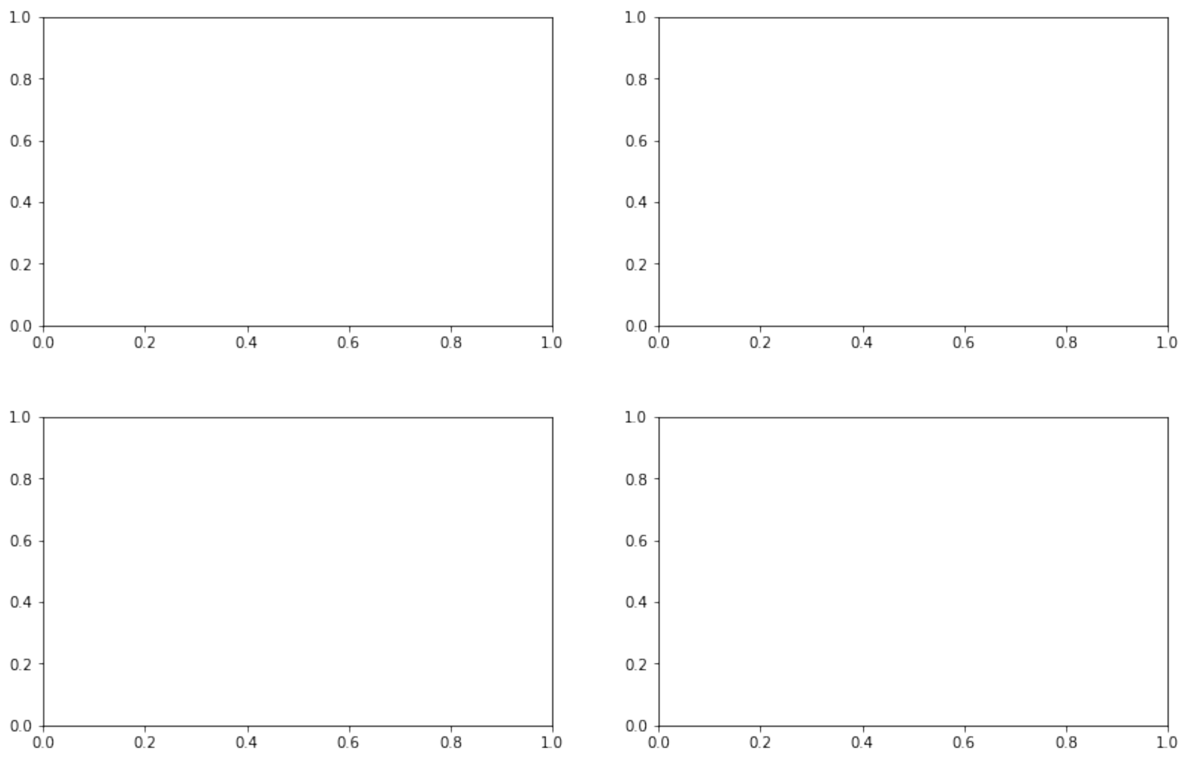
c. (shown below)

d. I can't seem to spot any trend by looking at the second plot.

e. (shown below)

f. Looking at the histogram we can observe that the sepal widths are approximately normally distributed among the 3 different species of iris flowers, meaning that this variable stays similar across species.

In [14]:
```
fig, axes = plt.subplots(2,2, figsize=(12,8))
fig.tight_layout(pad=4.0)
```



First Plot (a, b)

In [15]:
```
axes[0, 0].plot(iris.data)
axes[0,0].set_title("First Plot")
```

Out[15]:
```
Text(0.5, 1, 'First Plot')
```

Second plot (c, d)

In [16]:
```
axes[0,1].scatter(iris.data[:,0], iris.data[:,1])
axes[0,1].set_xlabel("sepal length (cm)")
axes[0,1].set_ylabel("sepal width (cm)")
axes[0,1].set_title("Second Plot")
```

Out[16]:
```
Text(0.5, 1, 'Second Plot')
```

Third plot (e)

In [17]:
```
every_5th_iris = iris.data[0::5]
```

In [18]:
```
axes[1,0].scatter(every_5th_iris[:,0], every_5th_iris[:,1])
axes[1,0].set_xlabel("sepal length (cm)")
axes[1,0].set_ylabel("sepal width (cm)")
axes[1,0].set_title("Third Plot")
```

Out[18]:
```
Text(0.5, 1, 'Third Plot')
```

Fourth plot (f)

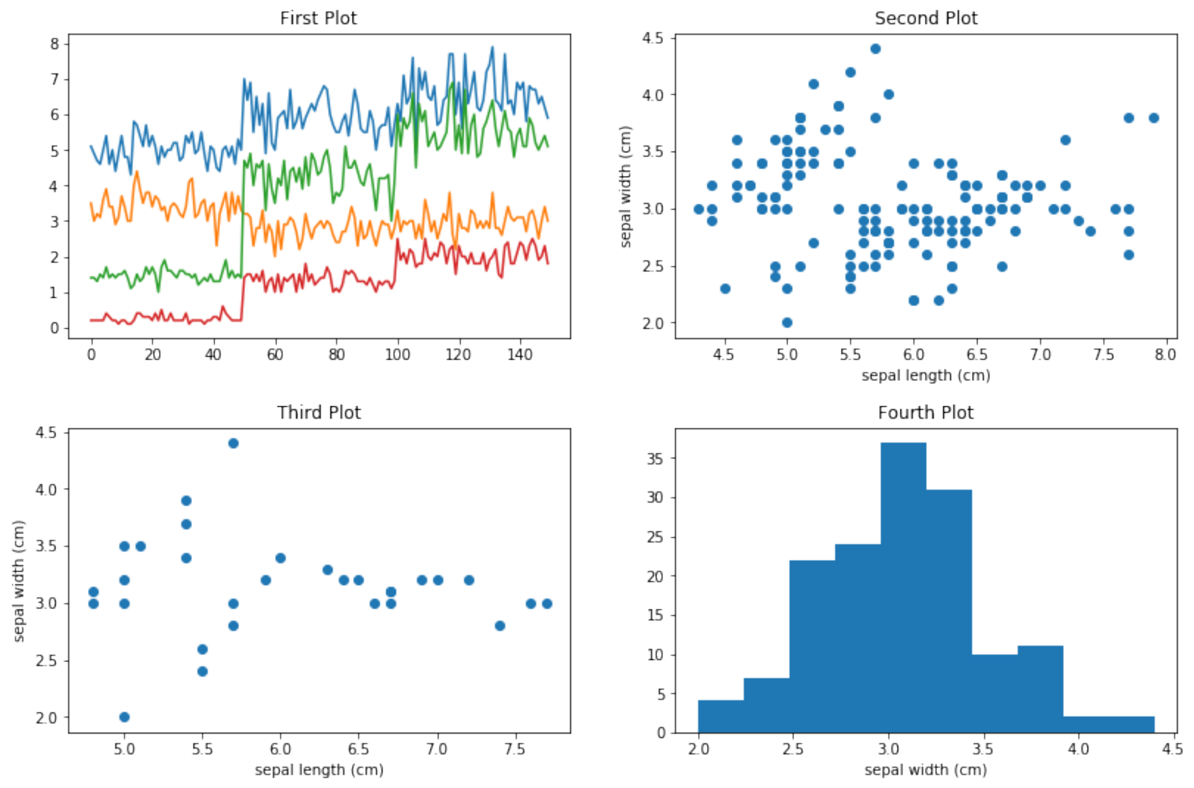
In [19]:
```
axes[1,1].hist(iris.data[:,1])
axes[1,1].set_xlabel('sepal width (cm)')
axes[1,1].set_title("Fourth Plot")
```

Out[19]:
```
Text(0.5, 1, 'Fourth Plot')
```

Subplots

In [20]:
```
fig
```

Out[20]:



6. Plotting with labels

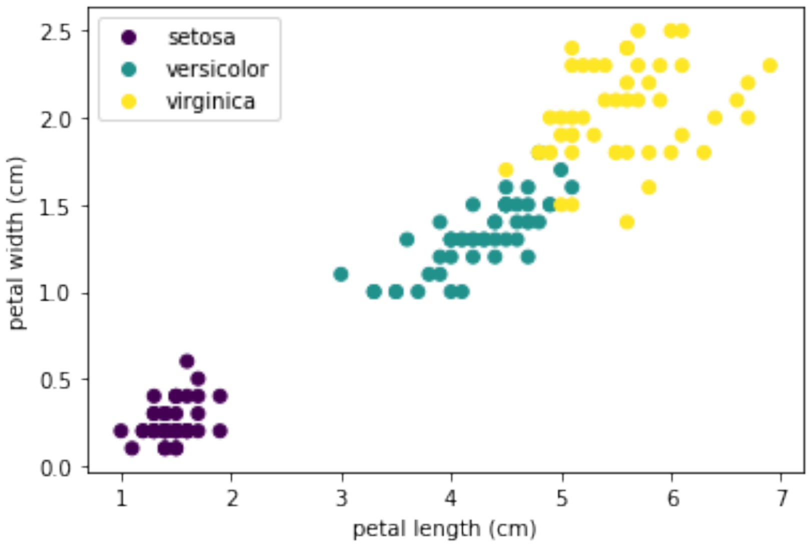
In [21]:
```
plt.scatter?
```

```
In [22]: data_obj = {'a': iris.data[:,2],
                     'b': iris.data[:,3],
                     'c': iris.target,
                    }

         plt.xlabel('petal length (cm)')
         plt.ylabel('petal width (cm)')

         sct = plt.scatter('a', 'b', c = 'c', data=data_obj) # Another way t
         o give x and y arrays by providing data
         plt.legend(handles=sct.legend_elements()[0],labels=['setosa', 'vers
         icolor', 'virginica'])

Out[22]: <matplotlib.legend.Legend at 0x10d40f750>
```



7. Video (Extra Credit)

https://drive.google.com/open?id=1g9WCt5NRHWhlc-GhvqzEpfsgHOwysU3I
(https://drive.google.com/open?id=1g9WCt5NRHWhlc-GhvqzEpfsgHOwysU3I)