Homework1

April 23, 2019

1 Data Mining Homework 1

Ross Lewis

1.1 Part I

1.1

The following program is used to sort the list according to the second element in the sublist. Please complete the missing lines.

```
In [4]: a=[['A',34],['B',21],['C',26]]
        for i in range(0,len(a)):
            for j in range(0,len(a)-i-1):
                if(a[j][1]>a[j+1][1]):
                    temp=a[j]
                    a[j] = a[j+1]
                    a[j+1] = temp
        print(a)
```

```
[['B', 21], ['C', 26], ['A', 34]]
```

1.2

The following program is used to clean up the string and remove all the punctuation marks. Please complete the missing lines.

```
In [7]: # define punctuation
        punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''

        my_str = "Hello!!!, he said ---and went."

        # To take input from the user
        # my_str = input("Enter a string: ")

        # remove punctuation from the string
        no_punct = ""
        for char in my_str:
            if(not char in punctuations):
```

```
            no_punct = no_punct + char

    # display the unpunctuated string
    print(no_punct)
```

```
Hello he said and went
```

1.3

Complete the following program to multiply two matrices using nested loops.

```
In [8]: sum([1,2])
```

```
Out[8]: 3
```

```
In [17]: # Program to multiply two matrices using nested loops

         # 3x3 matrix
         X = [[12,7,3],
             [4 ,5,6],
             [7 ,8,9]]
         # 3x4 matrix
         Y = [[5,8,1,2],
             [6,7,3,0],
             [4,5,9,1]]

         # result is 3x4
         result = [[0,0,0,0],
                   [0,0,0,0],
                   [0,0,0,0]]

         # iterate through rows of X
         for i in range(len(X)):
            # iterate through columns of Y
            for j in range(len(Y[0])):
                 # iterate through rows of Y
                 #for all values in the x row, multiply by the correspoding value in the Y col
                 runningTotal = 0
                 for index in range(len(X)):
                     runningTotal = runningTotal + X[i][index]*Y[index][j]
                 result[i][j] = runningTotal

         for r in result:
             print(r)
```

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

1.2 Part II

2.1

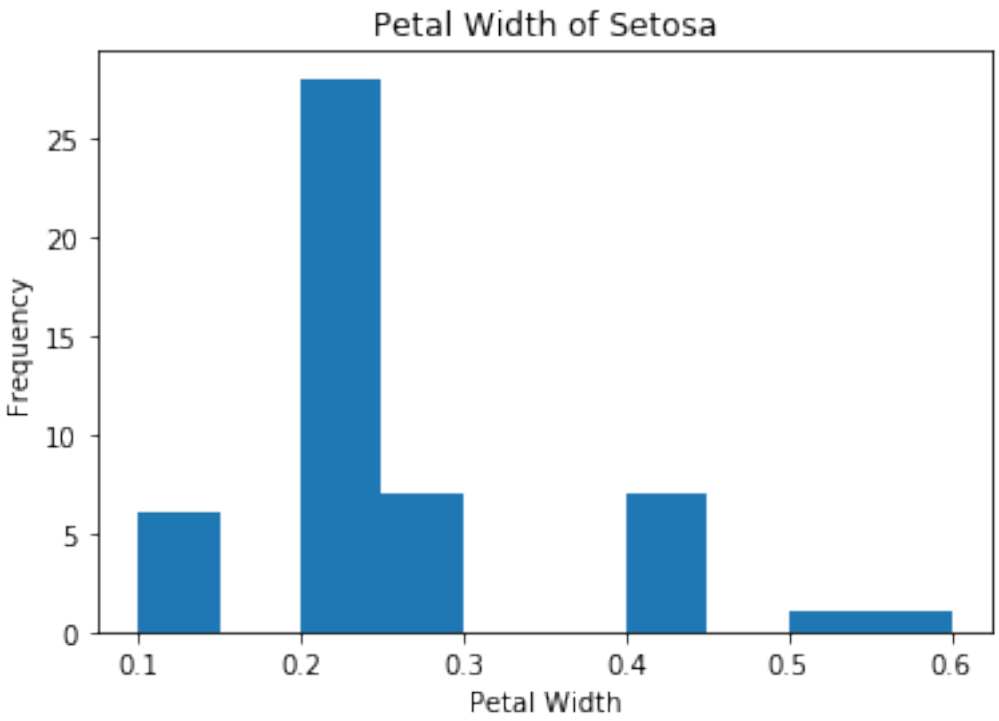
For each of the three flower categories, please quantize the petal width into 10 bins, and account the frequency of each bin, i.e., generating a 1-D histogram. Compare the three histograms side by side and explain your observations. You may use the function hist() or write your codes to implement the above objective.

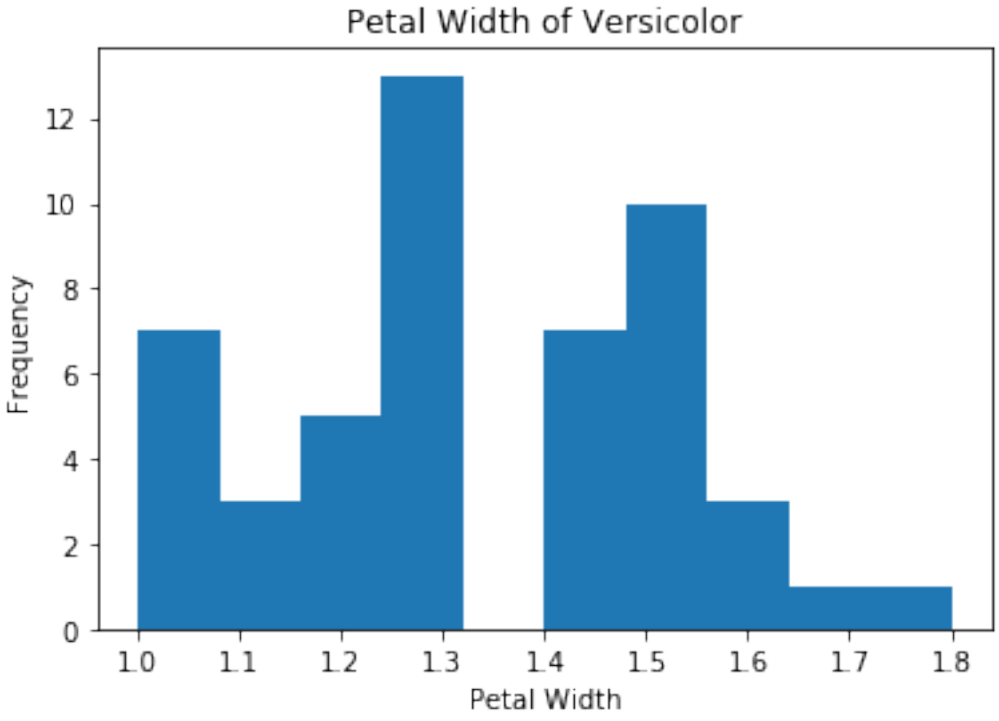
```
In [22]: import pandas
         iris = pandas.read_csv('iris.csv')

In [23]: iris.head()
```

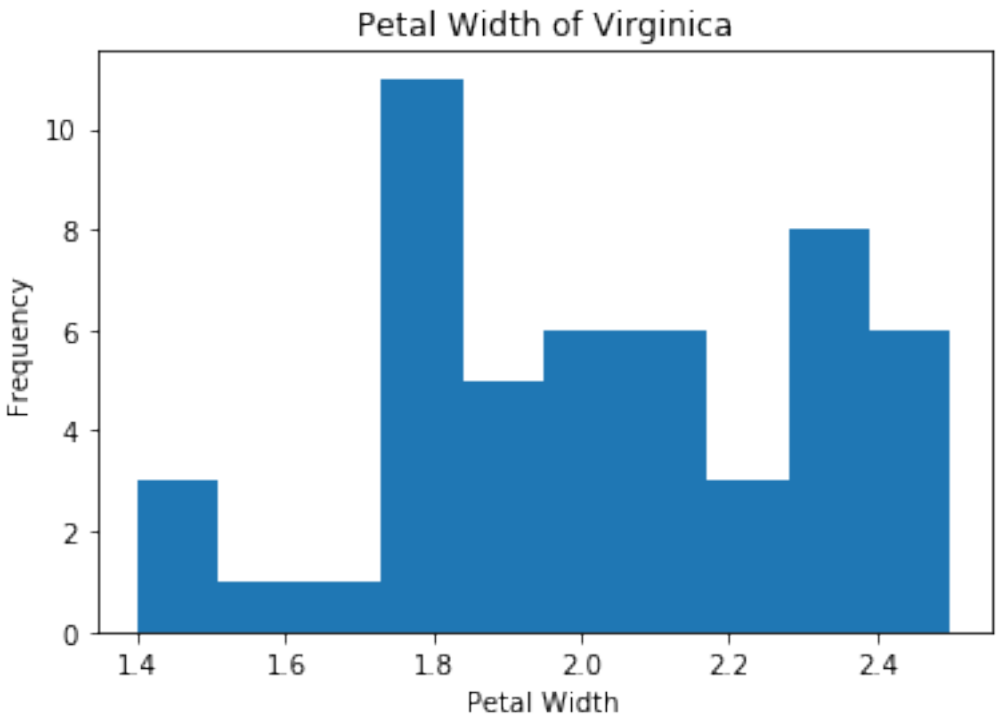
Out[23]:

	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

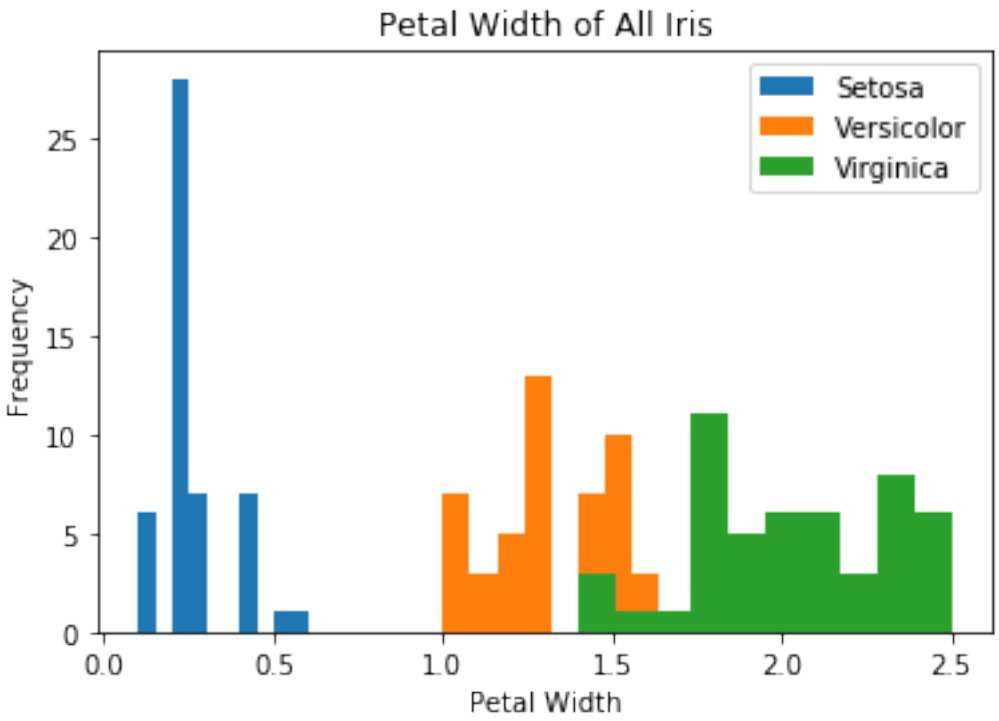
```
In [34]: iris['species'].unique()

Out[34]: array(['setosa', 'versicolor', 'virginica'], dtype=object)

In [46]: import matplotlib.pyplot as plt
         plt.hist(iris.where(iris['species'] == 'setosa')['petal_width'],bins=10)
         plt.title("Petal Width of Setosa")
         plt.xlabel("Petal Width")
         plt.ylabel("Frequency")
         plt.show()
         plt.hist(iris.where(iris['species'] == 'versicolor')['petal_width'],bins=10)
         plt.title("Petal Width of Versicolor")
         plt.xlabel("Petal Width")
         plt.ylabel("Frequency")
         plt.show()
         plt.hist(iris.where(iris['species'] == 'virginica')['petal_width'],bins=10)
         plt.title("Petal Width of Virginica")
         plt.xlabel("Petal Width")
         plt.ylabel("Frequency")
         plt.show()
```


Petal Width of Setosa
Frequency
0
5
10
15
20
25
0.1
0.2
0.3
0.4
0.5
0.6
Petal Width


Petal Width of Versicolor
Frequency
0
2
4
6
8
10
12
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
Petal Width




```
In [50]: plt.hist(iris.where(iris['species'] == 'setosa')['petal_width'],label='Setosa')
         plt.hist(iris.where(iris['species'] == 'versicolor')['petal_width'],label='Versicolor
         plt.hist(iris.where(iris['species'] == 'virginica')['petal_width'],label='Virginica')
         plt.title("Petal Width of All Iris")
         plt.xlabel("Petal Width")
         plt.ylabel("Frequency")
         plt.legend()
         plt.show()
```



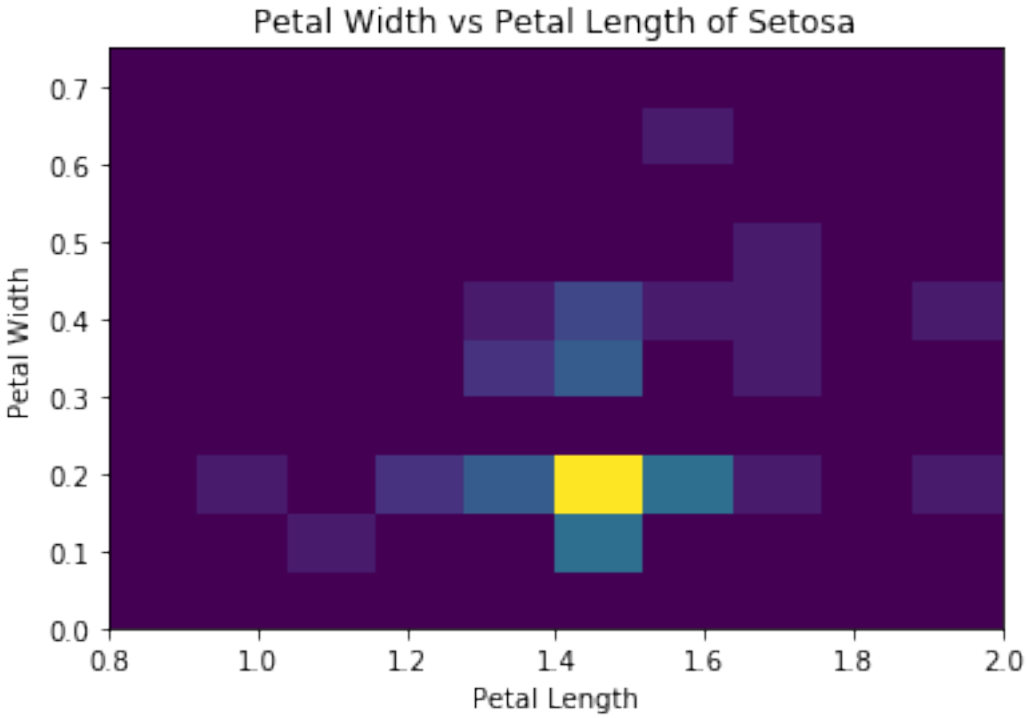
The setosa iris is the smallest of the three. Versicolor has a medium petal width, with some overlap with virginica. Virginica has the widest petal of the iris's being compared.

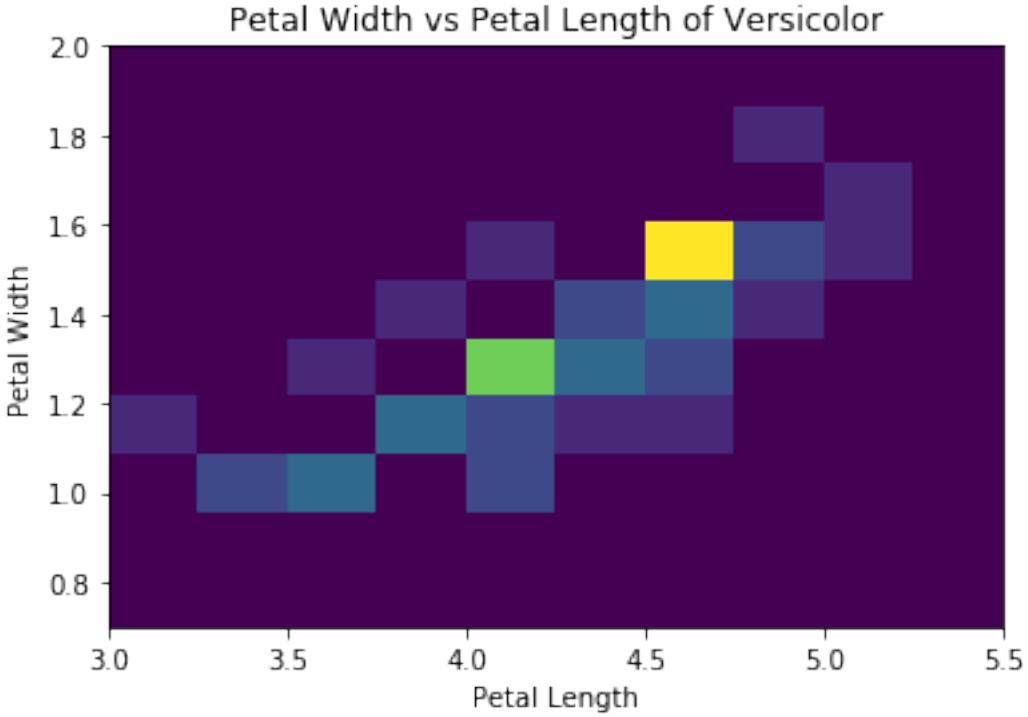
2.2

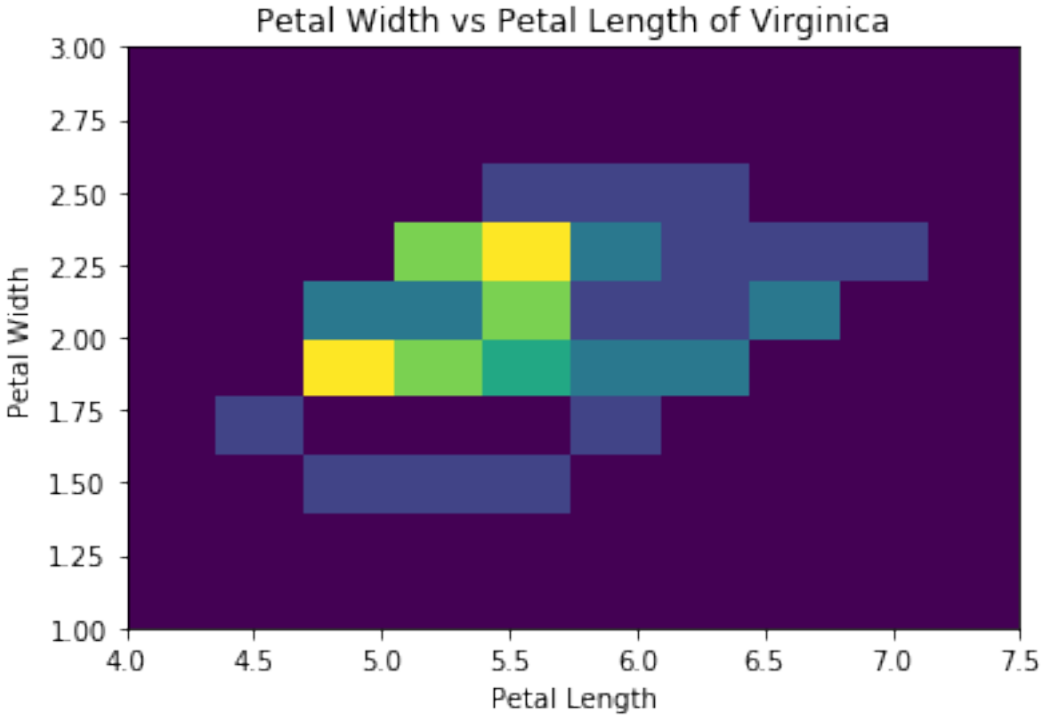
For each of the three categories, generate a 2D histogram

```
In [86]: plt.hist2d(x=iris.where(iris['species'] == 'setosa')['petal_length'].tolist(),
                    y=iris.where(iris['species'] == 'setosa')['petal_width'].tolist(),
                    range=[[.8,2],[0,.75]])
         plt.title("Petal Width vs Petal Length of Setosa")
         plt.xlabel("Petal Length")
         plt.ylabel("Petal Width")
         plt.show()
         plt.hist2d(x=iris.where(iris['species'] == 'versicolor')['petal_length'].tolist(),
                    y=iris.where(iris['species'] == 'versicolor')['petal_width'].tolist(),
                    range=[[3,5.5],[.7,2]])
         plt.title("Petal Width vs Petal Length of Versicolor")
         plt.xlabel("Petal Length")
         plt.ylabel("Petal Width")
         plt.show()
         plt.hist2d(x=iris.where(iris['species'] == 'virginica')['petal_length'].tolist(),
                    y=iris.where(iris['species'] == 'virginica')['petal_width'].tolist(),
                    range=[[4,7.5],[1,3]])
         plt.title("Petal Width vs Petal Length of Virginica")
         plt.xlabel("Petal Length")
```

```
plt.ylabel("Petal Width")
plt.show()
```





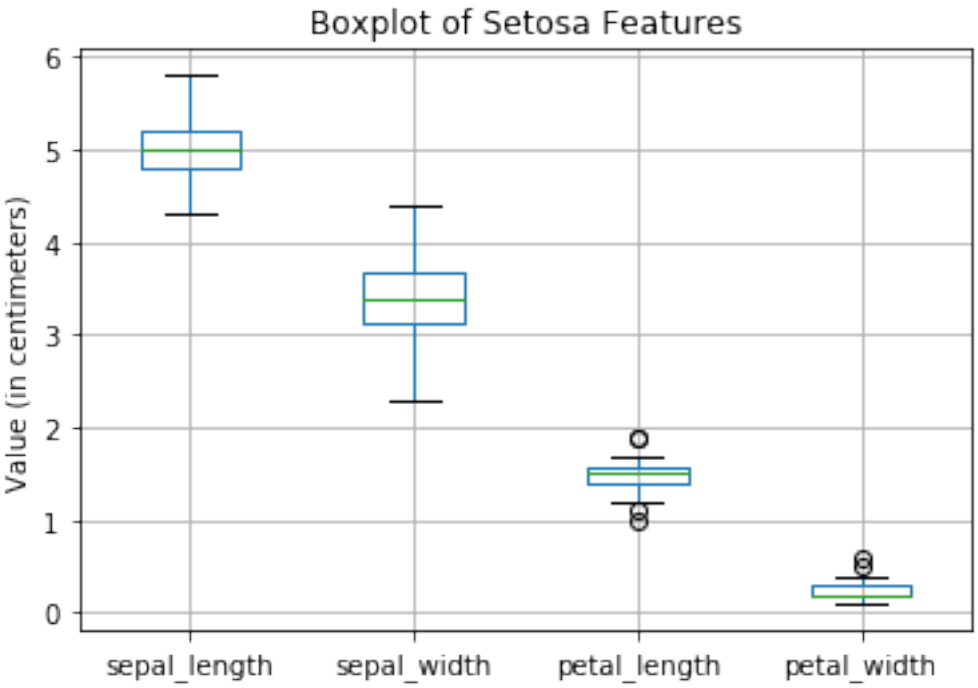


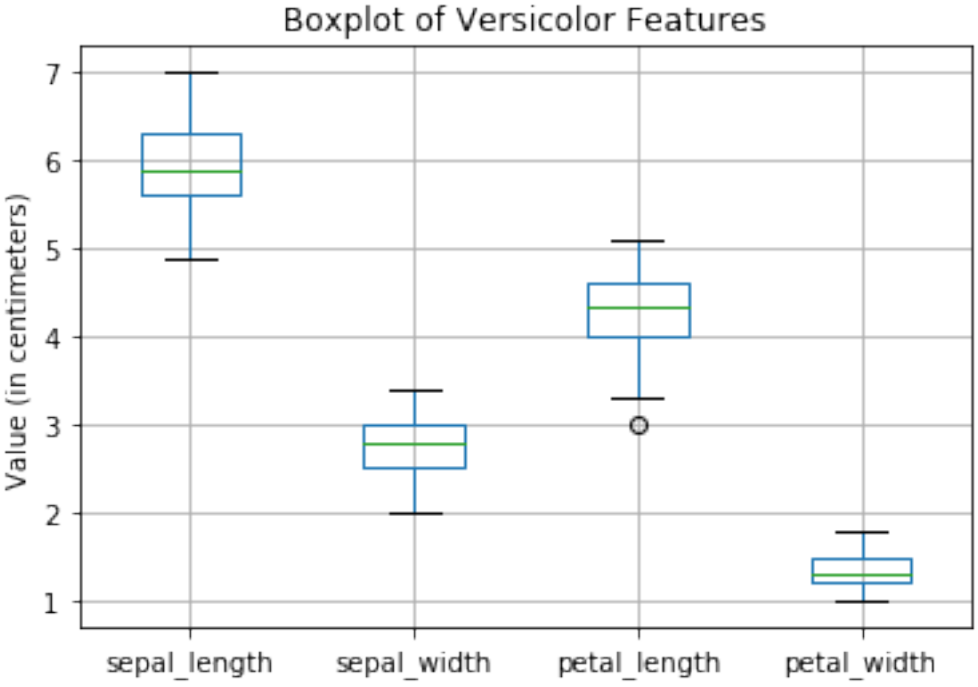
The range of size of the larger flowers tends to be wider. The axis labels are important for these, as we can see the Setosa's maximum petal length is shorter than the smallest Versicolor or Virginica.

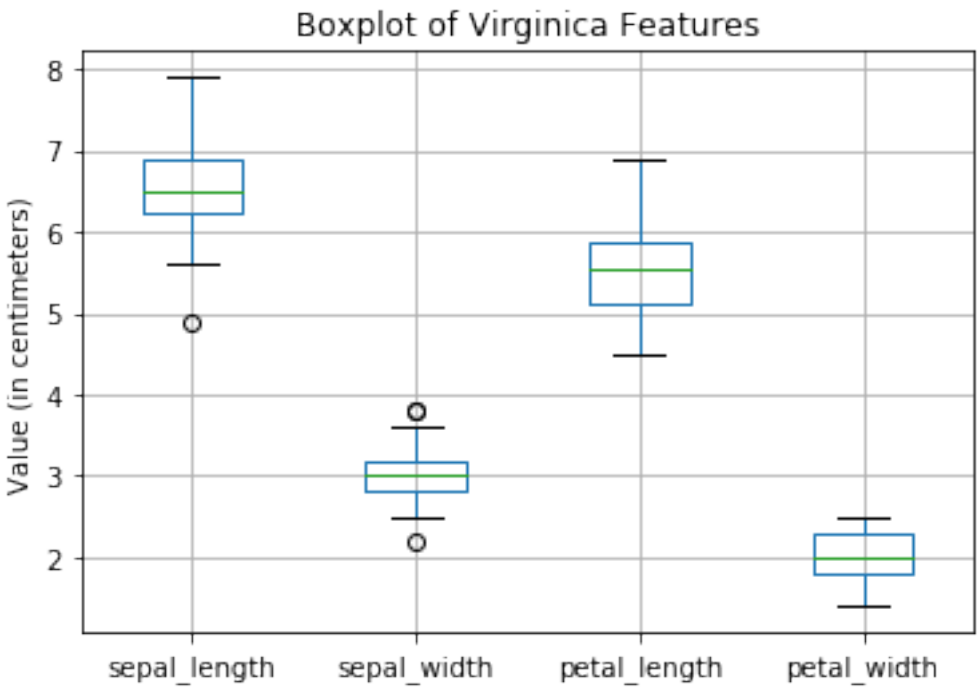
2.3

For each category, generate the BOX PLOTs for all four features

```
In [103]: iris.where(iris['species'] == 'setosa').boxplot(['sepal_length','sepal_width','petal_
          plt.title("Boxplot of Setosa Features")
          plt.ylabel("Value (in centimeters)")
          plt.show()
          iris.where(iris['species'] == 'versicolor').boxplot(['sepal_length','sepal_width','pe
          plt.title("Boxplot of Versicolor Features")
          plt.ylabel("Value (in centimeters)")
          plt.show()
          iris.where(iris['species'] == 'virginica').boxplot(['sepal_length','sepal_width','pet
          plt.title("Boxplot of Virginica Features")
          plt.ylabel("Value (in centimeters)")
          plt.show()
```


Boxplot of Setosa Features
Value (in centimeters)
0
1
2
3
4
5
6
sepal_length
sepal_width
petal_length
petal_width

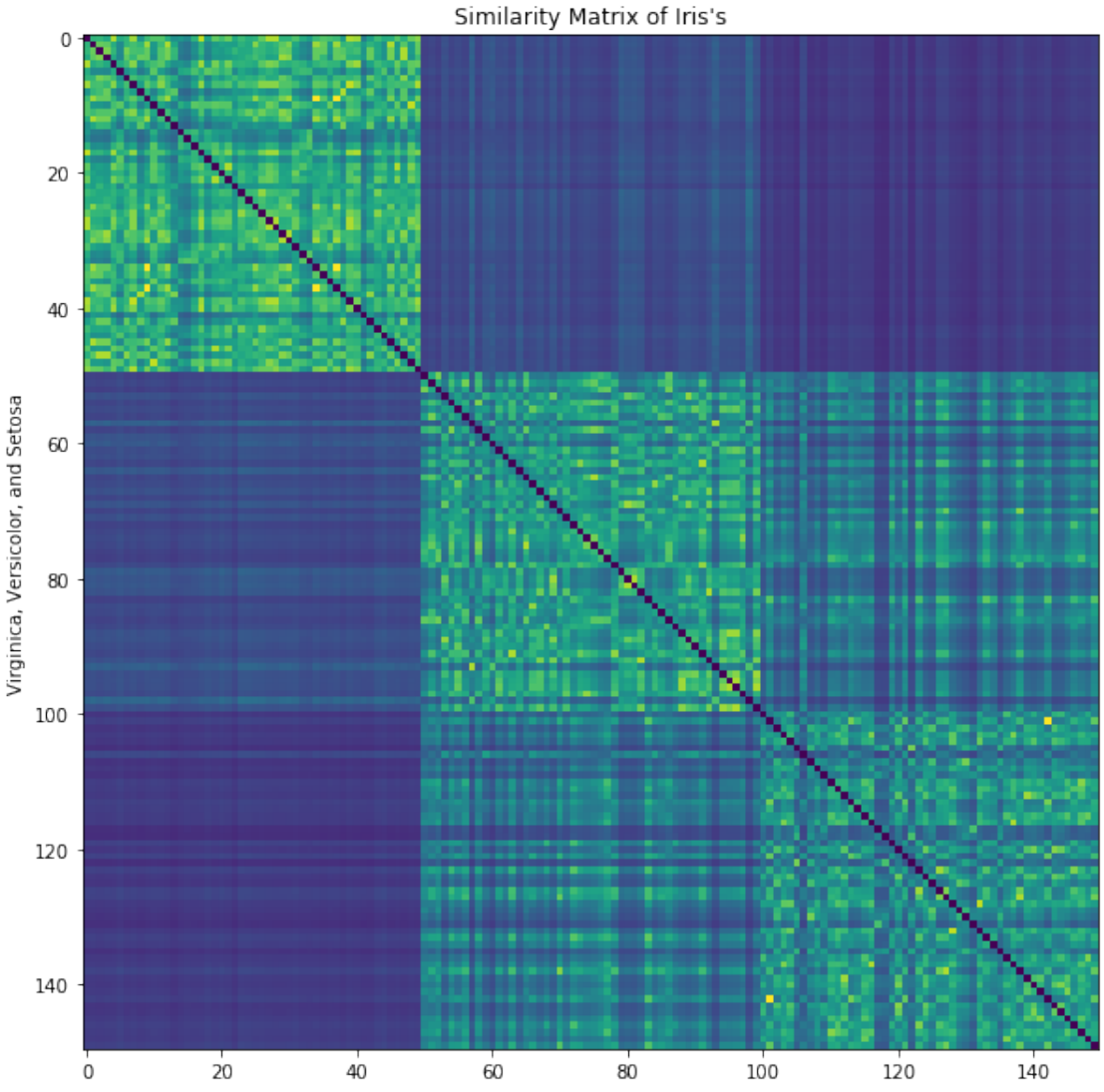

Boxplot of Versicolor Features
Value (in centimeters)
1
2
3
4
5
6
7
sepal_length
sepal_width
petal_length
petal_width



2.4

Visualization of similarity matrix.

```
In [114]: from scipy.spatial.distance import euclidean, pdist, squareform


          def similarity(u, v):
              return 1/(1+euclidean(u,v))

          dists = pdist(iris[['sepal_length','sepal_width','petal_length','petal_width']], sim
          euclid = pandas.DataFrame(squareform(dists), columns=iris.index)

In [121]: plt.figure(figsize=(10, 10))
          plt.imshow(euclid, interpolation='none')
          plt.title("Similarity Matrix of Iris's")
          plt.ylabel("Virginica, Versicolor, and Setosa")
          plt.show()
```


Similarity Matrix of Iris's
Virginica, Versicolor, and Setosa
0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140