Lab Work 02

January 31, 2018

```
In [1]: # Importing the library
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import matplotlib.patches as mp

        %matplotlib inline
```

1 Task 1:

Load the iris_dataset in your program in a python dataframe. View the First 5 and Last 3 row items of that dataframe.

```
In [2]: #Reading The Dataframe
        df = pd.read_csv('iris.csv')
```

```
In [3]: # Showning the first 5
        df.head(5)
```

```
Out[3]:    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 [4]: # Showning the last 3
        df.tail(3)
```

```
Out[4]:      sepal_length  sepal_width  petal_length  petal_width    species
        147           6.5          3.0           5.2          2.0  virginica
        148           6.2          3.4           5.4          2.3  virginica
        149           5.9          3.0           5.1          1.8  virginica
```

2 Task 2:

Show the data items in a sorted manner, sorted by 'Petal Length'.

```
In [5]: # Sorting by Petal Length and showing the first 5
        df.sort_values('petal_length').head()
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
22	4.6	3.6	1.0	0.2	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
35	5.0	3.2	1.2	0.2	setosa
36	5.5	3.5	1.3	0.2	setosa

3 Task 3:

Group data items by Species Name, calculate the mean for each of the 4 columns for each group, save them in a new dataframe and print

```
In [6]: # Saving new Datafram
        new_df = df.groupby(['species'][0:4])['sepal_length','sepal_width','petal_length','peta
        print(type(new_df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
In [7]: # Printing the Datafram
        print(new_df)
```

```
            sepal_length  sepal_width  petal_length  petal_width
species
setosa             5.006        3.418         1.464        0.244
versicolor         5.936        2.770         4.260        1.326
virginica          6.588        2.974         5.552        2.026
```

4 Task 4:

```
1) Plot Sepal_length VS Sepal_width in a Scatter Plot,
2) Petal_length VS Petal_width in another scatter plot.
3) Use subplot to plot this two plots in a single plot.
4) Different class points will have different color [Setosa = Red, Versicolor = Green, Virginic
```

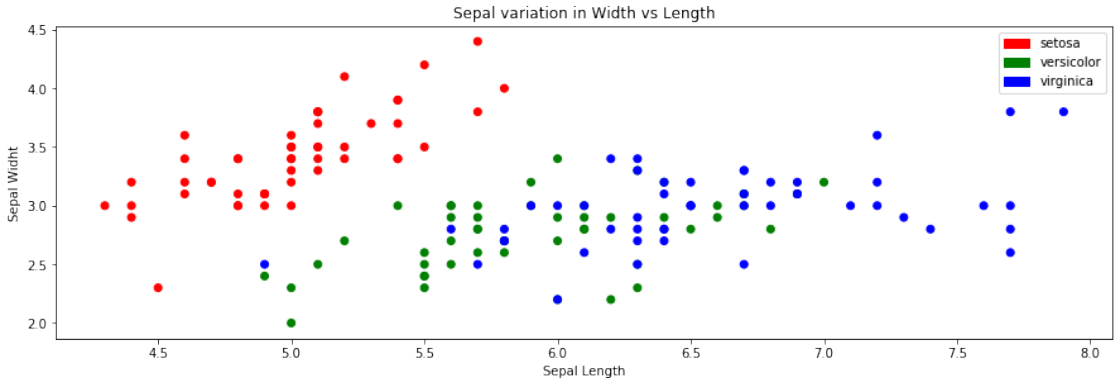
```
In [8]: # Assigning The Data
        sepal_length=df['sepal_length']
        sepal_width= df['sepal_width']
        petal_length = df['sepal_width']
        petal_width = df['petal_width']
        species = df['species']
```

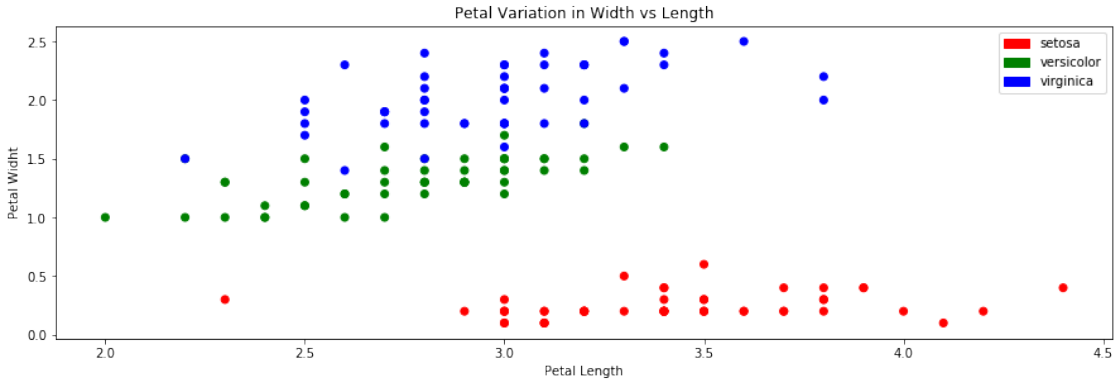
4.1 Subplot for Sepal_length VS Sepal_width and Petal_length VS Petal_width with Different class points will have different color

In [9]:
```
# create key, value pair dict to bind species to different colours
pairs={'setosa' :'r','versicolor' :'g','virginica' :'b'}
labels = [mp.Patch(color=cl, label=la) for la, cl in pairs.items()]

# Plot 01
plt.figure(figsize=(15,10))
plt.subplot(2, 1,2)
plt.scatter(sepal_length, sepal_width, c=[pairs[i] for i in species], label=[pairs[i] :
plt.ylabel('Sepal Widht')                          # set y label
plt.xlabel('Sepal Length')                         # set x label
plt.title('Sepal variation in Width vs Length')    # give it a title
plt.legend(handles = labels)


#Plot 02
plt.figure(figsize=(15,10))
plt.subplot(2, 1, 2)
plt.scatter(petal_length, petal_width, c=[pairs[i] for i in species], label=[pairs[i] :
plt.ylabel('Petal Widht')                          # set y label
plt.xlabel('Petal Length')                         # set x label
plt.title('Petal Variation in Width vs Length')    # give it a title
plt.legend(handles = labels)
plt.show()
```





5 Task 5:

1) Add a new column to the end of the dataframe, called 'Calyx Width'. If the flower's Sepal_le
2) Also plot the Histogram of 'Sepal Length' column.

5.1 Adding a new column to the end of the dataframe, called 'Calyx Width' and assigning the value

In [10]:
```
# function for checking sepal length
def check(x):
    y = []
    for i in range(len(x)):
        if df['sepal_length'][i] < 5:
            y.append(0)
        else:
            y.append(1)
    return y
```

In [11]:
```
# Method 01
df['Calyx Width'] =  check(df['sepal_length'])

# Methode 02
df['Calyx Width'] =  df.apply(lambda df: 0 if df['sepal_length'] < 5 else 1, axis=1)
```

In [12]:
```
# Showing the new Datafram
df.head()
```

Out[12]:

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

5.2 Ploting the Histogram of 'Sepal Length' column

```
In [13]: # ploting the Histogram
         plt.figure(figsize=(10,5))                    # set fig size
         plt.hist(sepal_length, bins=25,color='r')     # Set histogram
         plt.ylabel('Number')                          # set y label
         plt.xlabel('Sepal Length')                    # set x label
         plt.title('Sepal Length Histogram')           # give it a title
         plt.show()
```