

DATA EXPLORATION USING SEABORN

In [9]:

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

#Import the iris data set
from sklearn.datasets import load_iris
iris=load_iris()
# Assigning iris data set into data frame using features as column names
data=pd.DataFrame(iris.data,columns=iris.feature_names)
# creating label data frame with the coulms as Species types
label=pd.DataFrame(list(map(lambda x : iris.target_names[x],iris.target)),columns=['Species'])
iris=pd.concat([data,label],axis=1)
# Printin top 5 records of the iris data set
print(iris.head())
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
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	

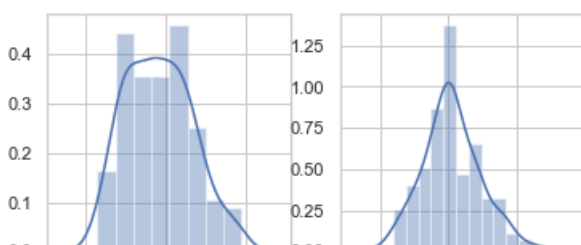
	Species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa

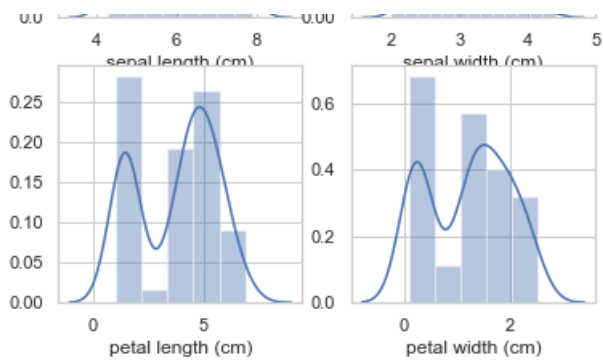
In [10]:

```
"""
1. Use distplot() to see the distribution of the Sepal length(cm), Sepal Width (cm), Petal length
(cm) and petal width
features. Plot them as subplots in single image
"""
```

```
"""
# Settin the style as white grid as background
sns.set_style("whitegrid");
# making fig size as 6X6 inches with 2 rows and 2 columns
fig, axs = plt.subplots(figsize=(6,6), ncols=2, nrows=2)
# displaying the graphs in each quadrant from 0,0 to 1,1
sns.distplot(iris['sepal length (cm)'],ax=axs[0, 0])
sns.distplot(iris['sepal width (cm)'], ax=axs[0, 1])
sns.distplot(iris['petal length (cm)'], ax=axs[1, 0])
sns.distplot(iris['petal width (cm)'], ax=axs[1, 1])
# displaying the Plot
plt.show();
```

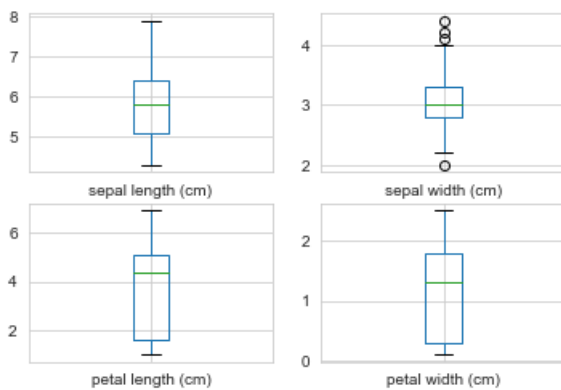
```
/home/chandu/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using
a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which
will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```





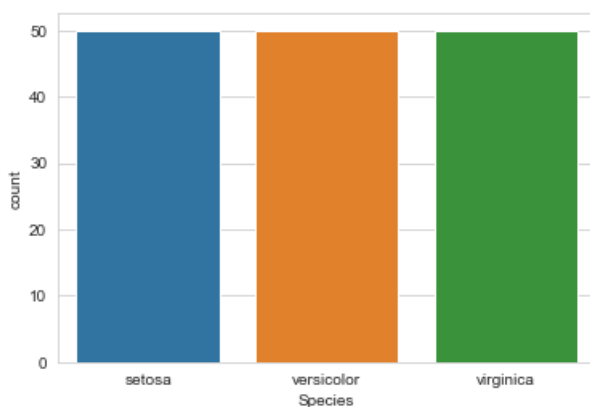
In [3]:

```
"""
2. Do a box plot for all the features except 'Species'
"""
# making figure and axes as 2 rows and 2 columns
fig, axes = plt.subplots(2,2)
# displaying plot for all the columns except Species which is last column(-1)
for i,el in enumerate(list(iris.columns.values)[: -1]):
    a = iris.boxplot(el, ax=axes.flatten()[i])
# displaying the Plot
plt.show()
```



In [4]:

```
"""
3. Do a countplot for feature Species
"""
# making count plot with Species column values
sns.countplot(x='Species',data=iris)
# displaying the Plot
plt.show()
```



In [5]:

```
"""
4. Do a pair plot on the features Sepal length(cm), Sepal Width (cm), Petal length (cm), petal width(cm) and Species
"""
```

```
plt.close();
sns.set_style("whitegrid");
# creating pair plots using Species as hue
sns.pairplot(iris, hue="Species", height=3);
# displaying the Plot
plt.show()
```

/home/chandu/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.

```
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [6]:

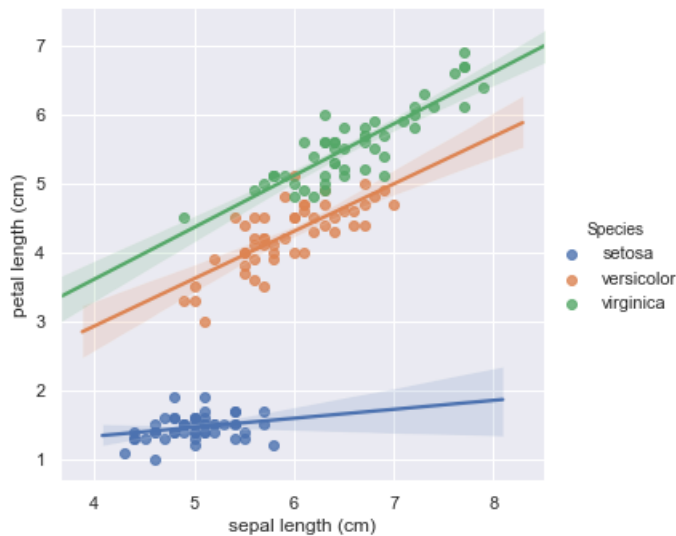
```
"""
5. Do Implot on the following Sepal length(cm), Petal length (cm). Using hue, display the differen
t speices in
different color
"""
```

```
plt.close();
sns.set_style("whitegrid");
# making color code as true to disply inbuilt color code
sns.set(color_codes=True)
# creating implot with sepal length and petal length using Species as hue
sns.Implot(x='sepal length (cm)',y='petal length (cm)',hue='Species',data=iris)
# displaying the Plot
plt.show()
```

```

/home/chandu/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using
a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which
will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

```



In [7]:

```

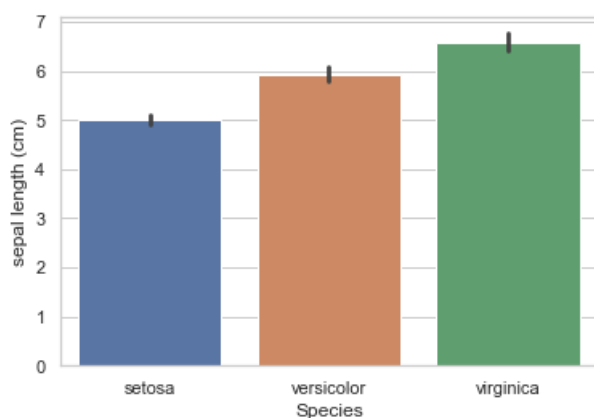
"""
6. Do a bar plot of Species vs sepal length (cm)
"""
# closing the existing plotting
plt.close();
# setting back ground style as white grid
sns.set_style("whitegrid");
# creating bar plot with species and sepal length
sns.barplot(x='Species',y='sepal length (cm)',data=iris)
# displaying the Plot
plt.show()

```

```

/home/chandu/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using
a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which
will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

```



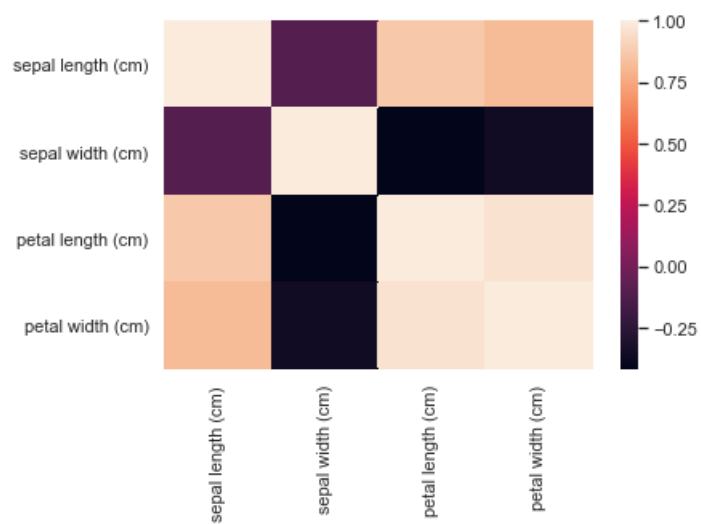
In [8]:

```

"""
7. Using heatmap, plot the correlation matrix
"""
# calculate the correlation matrix
corr = iris.corr()
# creating heatmap correlation column values
sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns)

```

```
# displaying the Plot  
plt.show()
```



In []: