DATA EXPLORATION USING SEABORN

In [1]:

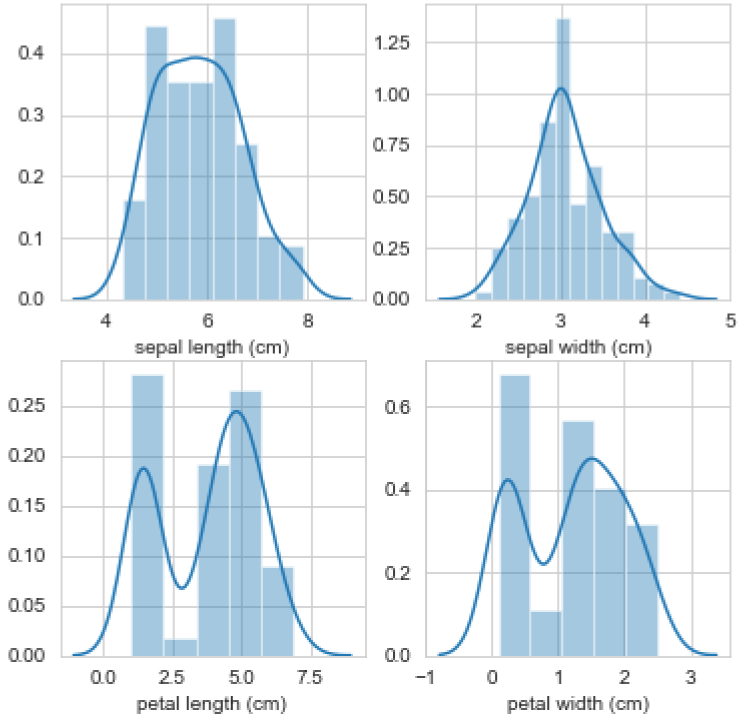
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

#Import the iris data set
from sklearn.datasets import load_iris
iris=load_iris()
data=pd.DataFrame(iris.data,columns=iris.feature_names)
label=pd.DataFrame(list(map(lambda x : iris.target_names[x],iris.target)),colu
iris=pd.concat([data,label],axis=1)
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 [2]:

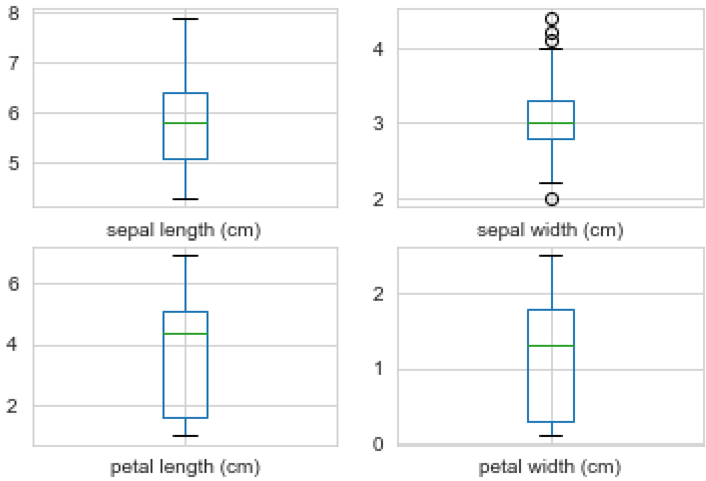
```
"""
1. Use distplot() to see the diistribution of the Sepal length(cm), Sepal Widt
features. Plot them as subplots in single image

"""
sns.set_style("whitegrid");
fig, axs = plt.subplots(figsize=(6,6), ncols=2, nrows=2)
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])
plt.show();
```

```
C:\Users\admin\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWa
rning: Using a non-tuple sequence for multidimensional indexing is deprecated;
use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interp
reted 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
```



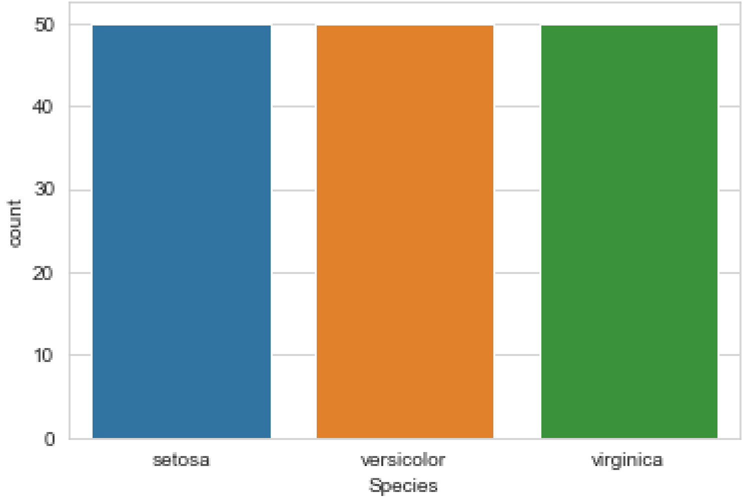
In [3]:

```
"""
2. Do a box plot for all the features except 'Species'

"""
fig, axes = plt.subplots(2,2)
for i,el in enumerate(list(iris.columns.values)[:-1]):
    a = iris.boxplot(el, ax=axes.flatten()[i])

plt.show()
```



In [4]:

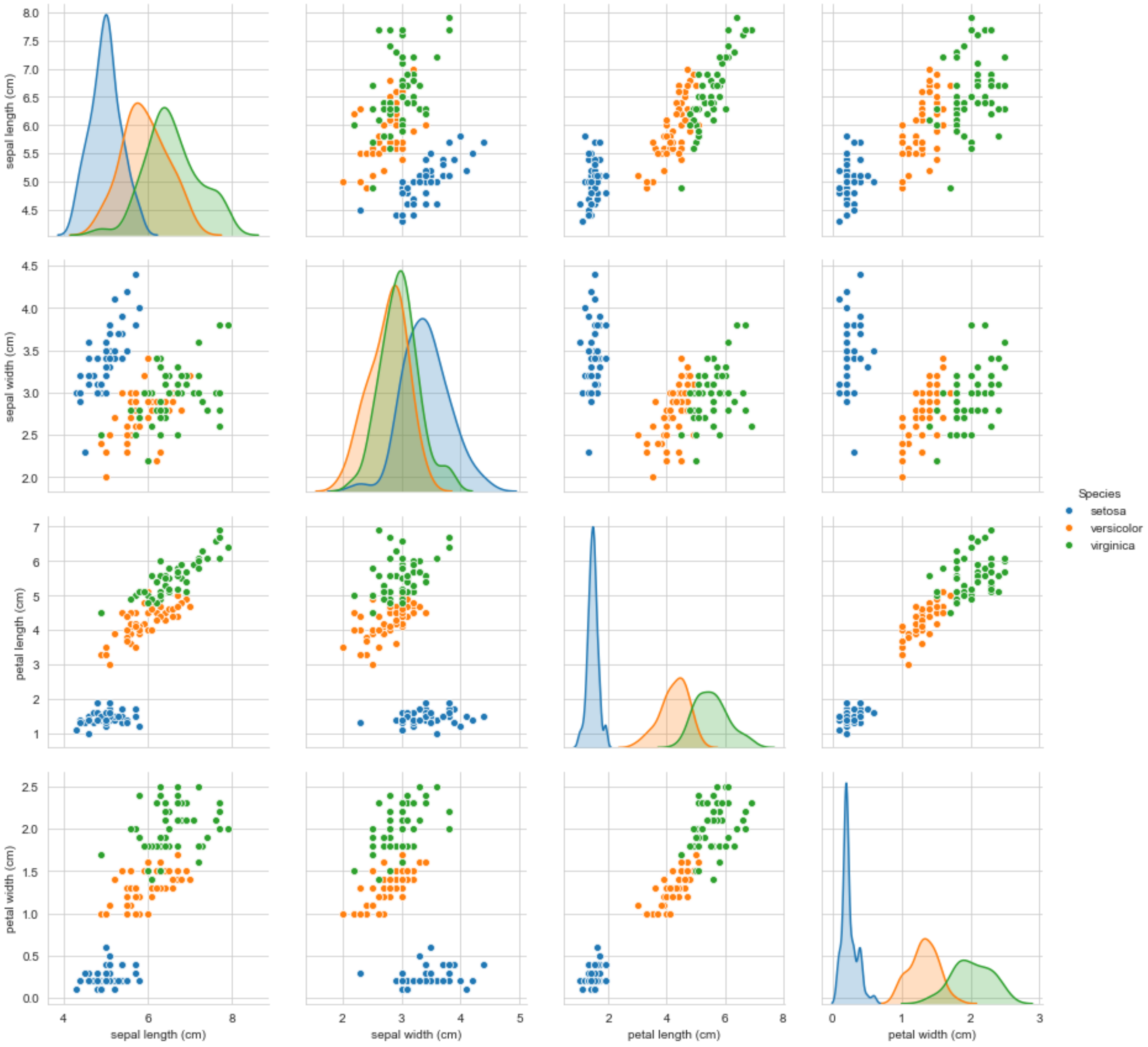
```
"""
3. Do a countplot for feature Species
"""
sns.countplot(x='Species',data=iris)
plt.show()
```



In [5]:

```
"""
4. Do a pair plot on the features Sepal length(cm), Sepal Width (cm), Petal le
"""

plt.close();
sns.set_style("whitegrid");
sns.pairplot(iris, hue="Species", height=3);
plt.show()
```

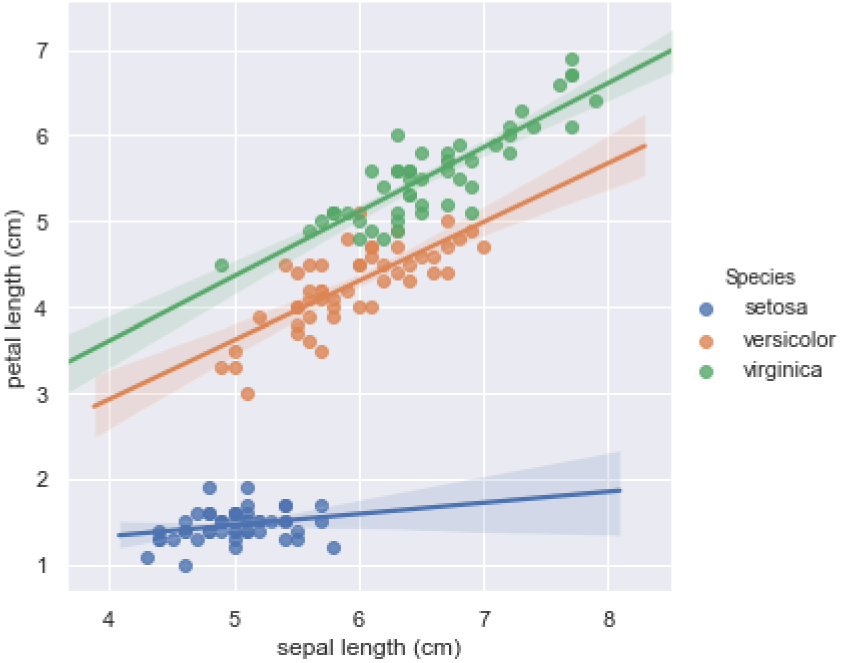
```
C:\Users\admin\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWa
rning: Using a non-tuple sequence for multidimensional indexing is deprecated;
use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interp
reted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [6]:

```
"""
5. Do lmplot on the following Sepal length(cm), Petal length (cm). Using hue,
different color
"""

plt.close();
sns.set_style("whitegrid");
sns.set(color_codes=True)
sns.lmplot(x='sepal length (cm)',y='petal length (cm)',hue='Species',data=iris
plt.show()

```

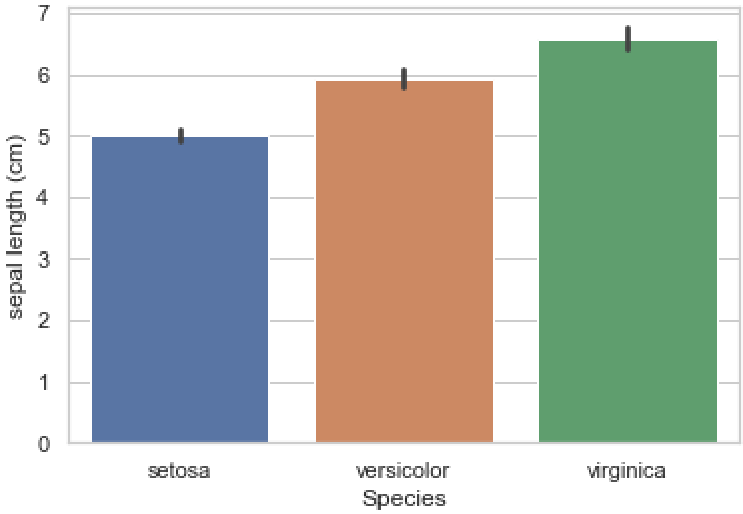
```
C:\Users\admin\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWa
rning: Using a non-tuple sequence for multidimensional indexing is deprecated;
use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interp
reted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



In [7]:

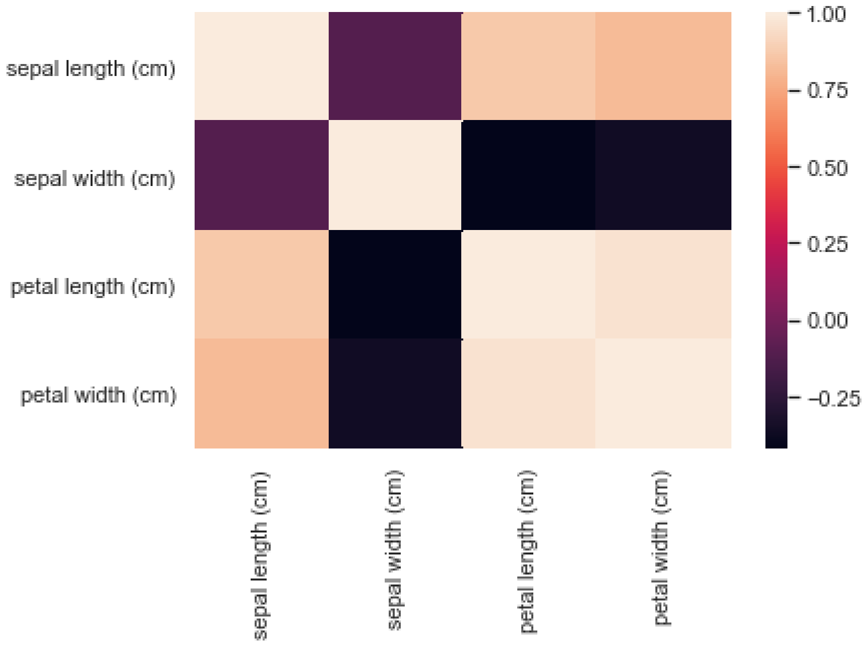
```
"""
6. Do a bar plot of Species vs sepal length (cm)
"""
plt.close();
sns.set_style("whitegrid");
sns.barplot(x='Species',y='sepal length (cm)',data=iris)
plt.show()
```

```
C:\Users\admin\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWa
rning: Using a non-tuple sequence for multidimensional indexing is deprecated;
use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interp
reted 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()

sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns)
plt.show()
```



In [ ]:

```

```