In [1]:

```
from sklearn.datasets import load_iris
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv(
    filepath_or_buffer='https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',
    header=None,
    sep=',')

df.columns=['sepal_len', 'sepal_wid', 'petal_len', 'petal_wid', 'class']
df.dropna(how="all", inplace=True) # drops the empty line at file-end

df.tail()
```

Out[2]:

	sepal_len	sepal_wid	petal_len	petal_wid	class
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

In [3]:

```
df.shape
```

Out[3]:

```
(150, 5)
```

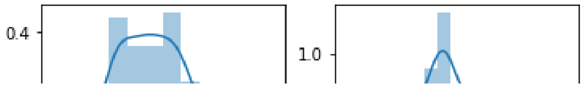
In [19]:

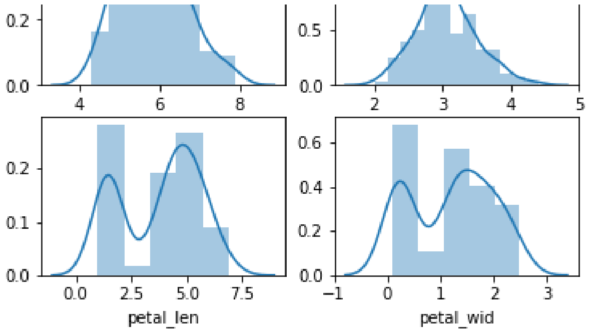
```
fig, axs = plt.subplots(ncols=2,nrows=2)
sns.distplot(df['sepal_len'],kde=True,ax=axs[0,0])
sns.distplot(df['sepal_wid'],kde=True,ax=axs[0,1])
sns.distplot(df['petal_len'],kde=True,ax=axs[1,0])
sns.distplot(df['petal_wid'],kde=True,ax=axs[1,1])
```

```
/home/yeshua/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The
'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/home/yeshua/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The
'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/home/yeshua/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The
'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/home/yeshua/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: The
'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7a51700b8>
```



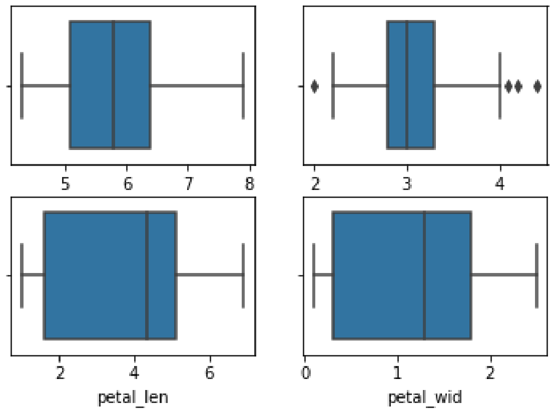


In [22]:

```
fig, axs = plt.subplots(ncols=2,nrows=2)
sns.boxplot(df['sepal_len'],ax=axs[0,0],orient='h')
sns.boxplot(df['sepal_wid'],ax=axs[0,1],orient='h')
sns.boxplot(df['petal_len'],ax=axs[1,0],orient='h')
sns.boxplot(df['petal_wid'],ax=axs[1,1],orient='h')
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7a5331400>
```

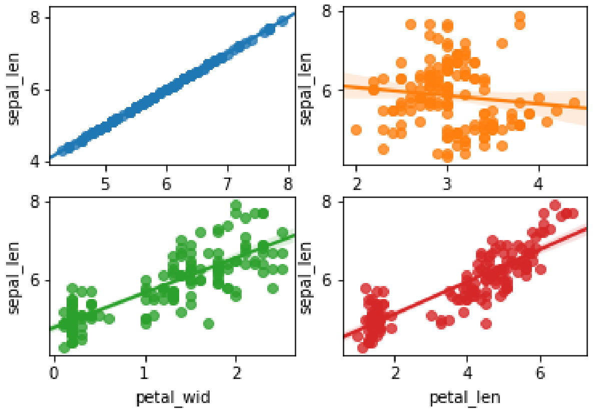


In [23]:

```
fig, axs = plt.subplots(ncols=2,nrows=2)
sns.regplot(x='sepal_len',y='sepal_len',data=df,ax=axs[0,0])
sns.regplot(x='sepal_wid',y='sepal_len',data=df,ax=axs[0,1])
sns.regplot(x='petal_wid',y='sepal_len',data=df,ax=axs[1,0])
sns.regplot(x='petal_len',y='sepal_len',data=df,ax=axs[1,1])
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe7a53ab2b0>
```



In [7]:

```
type('iris.data')
type('iris.target')
```

Out[7]:

```
str
```

In [8]:

```
iris.data.shape
```

Out[8]:

```
(150, 4)
```

In [9]:

```
iris.target.shape
```

Out[9]:

```
(150,)
```

In [10]:

```
featuresAll=[]
features = iris.data[: , [0,1,2,3]]
features.shape
```

Out[10]:

```
(150, 4)
```

In [11]:

```
targets = iris.target
targets.reshape(targets.shape[0],-1)
targets.shape
```

Out[11]:

```
(150,)
```

In [12]:

```
for observation in features:
    featuresAll.append([observation[0] + observation[1] + observation[2] + observation[3]])
print (featuresAll)
```

```
[[10.2], [9.5], [9.4], [9.399999999999999], [10.2], [11.4], [9.700000000000001], [10.1], [8.9], [9
.6], [10.8], [9.999999999999998], [9.299999999999999], [8.5], [11.2], [12.000000000000002],
[11.000000000000002], [10.3], [11.5], [10.7], [10.7], [10.700000000000001], [9.399999999999999], [
10.599999999999998], [10.299999999999999], [9.799999999999999], [10.4], [10.399999999999999], [10.
2], [9.7], [9.7], [10.700000000000001], [10.9], [11.299999999999999], [9.6], [9.599999999999998],
[10.5], [9.6], [8.9], [10.2], [10.100000000000001], [8.4], [9.1], [10.7], [11.2], [9.5],
[10.699999999999998], [9.399999999999999], [10.7], [9.9], [16.299999999999997],
[15.600000000000001], [16.4], [13.100000000000001], [15.4], [14.3], [15.9], [11.600000000000001],
[15.4], [13.200000000000001], [11.5], [14.600000000000001], [13.2], [15.1], [13.4],
[15.600000000000001], [14.6], [13.6], [14.4], [13.1], [15.700000000000003], [14.2],
[15.200000000000001], [14.799999999999997], [14.900000000000002], [15.4], [15.799999999999999], [1
6.4], [14.9], [12.8], [12.799999999999999], [12.600000000000001], [13.6], [15.399999999999999], [1
4.4], [15.5], [16.0], [14.3], [14.0], [13.3], [13.7], [15.1], [13.6], [11.6], [13.8],
[14.099999999999998], [14.100000000000001], [14.7], [11.7], [13.9], [18.1], [15.5], [18.1],
[16.599999999999998], [17.5], [19.3], [13.6], [18.3], [16.8], [19.4], [16.799999999999997],
[16.3], [17.400000000000002], [15.2], [16.099999999999998], [17.200000000000003], [16.8], [20.4],
[19.500000000000004], [14.7], [18.1], [15.299999999999999], [19.2], [15.700000000000001], [17.8],
[18.2], [15.600000000000001], [15.8], [16.9], [17.6], [18.199999999999996], [20.1], [17.0],
[15.7], [15.7], [19.099999999999998], [17.7], [16.8], [15.600000000000001], [17.5], [17.8],
[17.4], [15.5], [18.2], [18.2], [17.2], [15.700000000000001], [16.7], [17.3], [15.8]]
```

In [13]:

```
import matplotlib.pyplot as plt
```

```
plt.scatter(featuresAll, targets, color='red', alpha =1.0)
plt.rcParams['figure.figsize'] = [10,8]
plt.title('Iris Dataset scatter Plot')
plt.xlabel('Features')
plt.ylabel('Targets')
```

Out[13]:

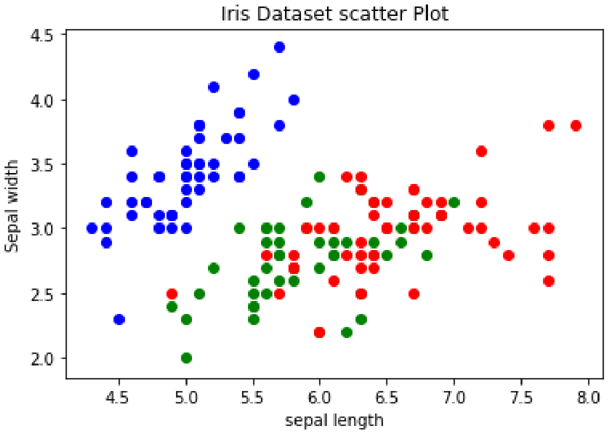
```
Text(0,0.5,'Targets')
```

In [14]:

```
featuresAll = []
targets = []
for feature in features:
    featuresAll.append(feature[0]) #Sepal length
    targets.append(feature[1]) #sepal width

groups = ('Iris-setosa','Iris-versicolor','Iris-virginica')
colors = ('blue', 'green','red')
data = ((featuresAll[:50], targets[:50]), (featuresAll[50:100], targets[50:100]),
        (featuresAll[100:150], targets[100:150]))

for item, color, group in zip(data,colors,groups):
    #item = (featuresAll[:50], targets[:50]), (featuresAll[50:100], targets[50:100]),
(featuresAll[100:150], targets[100:150])
    x, y = item
    plt.scatter(x, y,color=color,alpha=1)
    plt.title('Iris Dataset scatter Plot')
plt.xlabel('sepal length')
plt.ylabel('Sepal width')
plt.show()
```



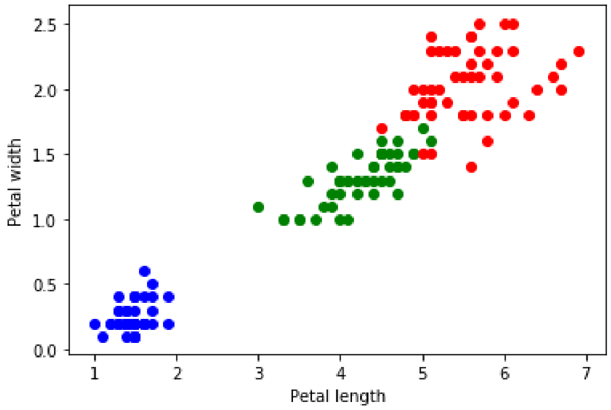
In [15]:

```
featuresAll = []
targets = []
for feature in features:
    featuresAll.append(feature[2]) #Petal length
    targets.append(feature[3]) #Petal width

groups = ('Iris-setosa','Iris-versicolor','Iris-virginica')
colors = ('blue', 'green','red')
data = ((featuresAll[:50], targets[:50]), (featuresAll[50:100], targets[50:100]),
        (featuresAll[100:150], targets[100:150]))

for item, color, group in zip(data,colors,groups):
    #item = (featuresAll[:50], targets[:50]), (featuresAll[50:100], targets[50:100]),
(featuresAll[100:150], targets[100:150])
    x0, y0 = item
    plt.scatter(x0, y0,color=color,alpha=1)
    plt.title('Iris Dataset scatter Plot')
plt.xlabel('Petal length')
plt.ylabel('Petal width')
plt.show()
```

Iris Dataset scatter Plot



In [16]:

```
import pandas as pd
iris = load_iris()
ir = pd.DataFrame(iris.data)
ir.columns = iris.feature_names
ir['CLASS'] = iris.target
ir.head()
```

Out[16]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	CLASS
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [17]:

```
from sklearn.neighbors import NearestNeighbors
nn = NearestNeighbors(5) #The arguements specify to return the Fast 5 most among the dataset
nn.fit(iris.data)
```

Out[17]:

```
NearestNeighbors(algorithm='auto', leaf_size=30, metric='minkowski',
         metric_params=None, n_jobs=1, n_neighbors=5, p=2, radius=1.0)
```

In [18]:

```
ir.describe()
```

Out[18]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	CLASS
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667	1.000000
std	0.828066	0.433594	1.764420	0.763161	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

In [19]:

```
import numpy as np
test = np.array([5.4,2,2,2.3])
test1 = test.reshape(1,-1)
test1.shape
```

Out[19]:

```
(1, 4)
```

In [20]:

```
nn.kneighbors(test1,5)
```

Out[20]:

```
(array([[1.6673332 , 1.90525589, 1.94679223, 2.02484567, 2.09523268]]),
 array([[98, 93, 57, 60, 79]], dtype=int64))
```

In [21]:

```
ir.ix[[98, 93, 57, 60, 79],]
```

```
E:\sreeproject\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#ix-indexer-is-deprecated
  """Entry point for launching an IPython kernel.
```

Out[21]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	CLASS
98	5.1	2.5	3.0	1.1	1
93	5.0	2.3	3.3	1.0	1
57	4.9	2.4	3.3	1.0	1
60	5.0	2.0	3.5	1.0	1
79	5.7	2.6	3.5	1.0	1