In [1]:
```
#identifying the versions
import sys
print('Python:{}'.format(sys.version))
import scipy
print('Scipy:{}'.format(scipy.__version__))
import numpy
print('Numpy:{}'.format(numpy.__version__))
import matplotlib
print('Matplotlib:{}'.format(matplotlib.__version__))
import pandas
print('Pandas:{}'.format(pandas.__version__))
import sklearn
print('Sklearn:{}'.format(sklearn.__version__))
```

```
Python:3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Scipy:1.4.1
Numpy:1.18.1
Matplotlib:3.1.3
Pandas:1.0.1
Sklearn:0.22.1
```

In [2]:
```
import pandas as pd
import numpy as np
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn import datasets,linear_model
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import model_selection
from sklearn.ensemble import VotingClassifier
```

In [3]:
```
#loading the dataset
names=['sepal-length','sepal-width','petal-length','class']
dataset=read_csv('iris.csv',names=names)
```

In [4]:
```
#dimension of the dataset
print(dataset.shape)
```

```
(151, 4)
```

In [5]:
```
#take a peek at the data
print(dataset.head(20))
```

```
             sepal-length   sepal-width petal-length        class
sepal_length  sepal_width  petal_length  petal_width      species
5.1                   3.5           1.4          0.2  Iris-setosa
4.9                     3           1.4          0.2  Iris-setosa
4.7                   3.2           1.3          0.2  Iris-setosa
4.6                   3.1           1.5          0.2  Iris-setosa
5                     3.6           1.4          0.2  Iris-setosa
5.4                   3.9           1.7          0.4  Iris-setosa
4.6                   3.4           1.4          0.3  Iris-setosa
5                     3.4           1.5          0.2  Iris-setosa
4.4                   2.9           1.4          0.2  Iris-setosa
4.9                   3.1           1.5          0.1  Iris-setosa
5.4                   3.7           1.5          0.2  Iris-setosa
4.8                   3.4           1.6          0.2  Iris-setosa
4.8                     3           1.4          0.1  Iris-setosa
4.3                     3           1.1          0.1  Iris-setosa
5.8                     4           1.2          0.2  Iris-setosa
5.7                   4.4           1.5          0.4  Iris-setosa
5.4                   3.9           1.3          0.4  Iris-setosa
5.1                   3.5           1.4          0.3  Iris-setosa
5.7                   3.8           1.7          0.3  Iris-setosa
```

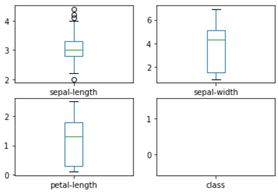
In [6]:
```
#statistical summary
print(dataset.describe())
```

```
       sepal-length sepal-width petal-length            class
count           151         151          151              151
unique           24          44           23                4
top               3         1.5          0.2  Iris-versicolor
freq             26          14           28               50
```

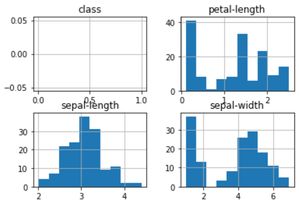
In [7]:
```
#class distribution
print(dataset.groupby('class').size())
```

```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
species             1
dtype: int64
```

In [8]:
```
#univariate plots-box and whisker plots
dataset = dataset.apply( pd.to_numeric, errors='coerce' )
dataset.plot(kind='box',subplots=True,layout=(2,2),sharex=False,sharey=False)
pyplot.show()
```



In [9]:
```
#histogram of the variable
dataset.hist()
pyplot.show()
```



In [10]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
%matplotlib inline

iris = load_iris()
colors = list()
palette = {0: "red", 1: "green", 2: "blue"}

for c in np.nditer(iris.target): colors.append(palette[int(c)])
    # using the palette dictionary, we convert
    # each numeric class into a color string
dataframe = pd.DataFrame(iris.data,
columns=iris.feature_names)
scatterplot = pd.plotting.scatter_matrix(dataframe, alpha=0.3,
figsize=(10, 10), diagonal='hist', c=colors, marker='o', grid=True)
```