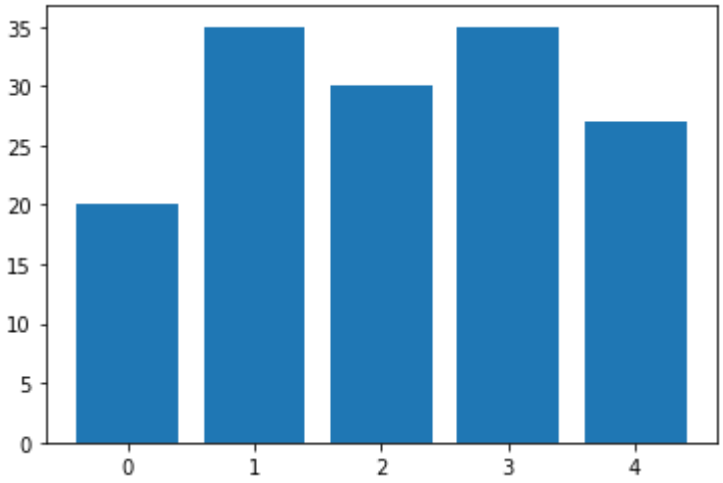
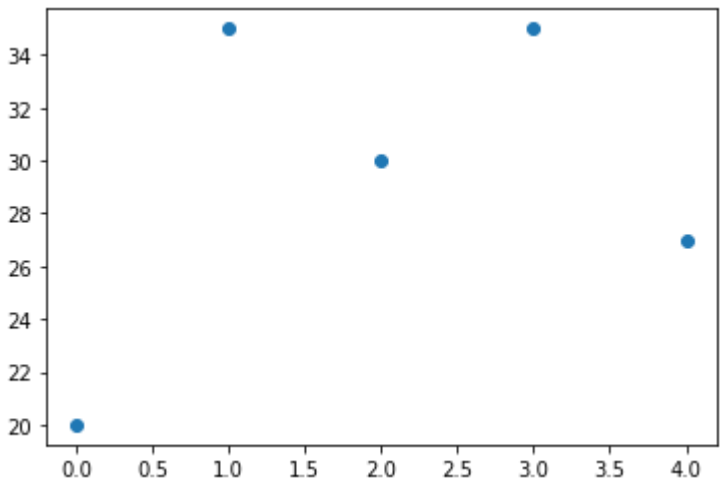
In [1]:

```
# simple bar and scatter plot
import numpy as np
from matplotlib import pyplot as plt
x = np.arange(5) # assume there are 5 students
y = (20, 35, 30, 35, 27) # their test scores
plt.bar(x,y) # Bar plot
# need to close the figure using show() or close(), if not closed any follow
#up plot commands will use same figure.
plt.show() # Try commenting this an run
plt.scatter(x,y) # scatter plot
plt.show()
```



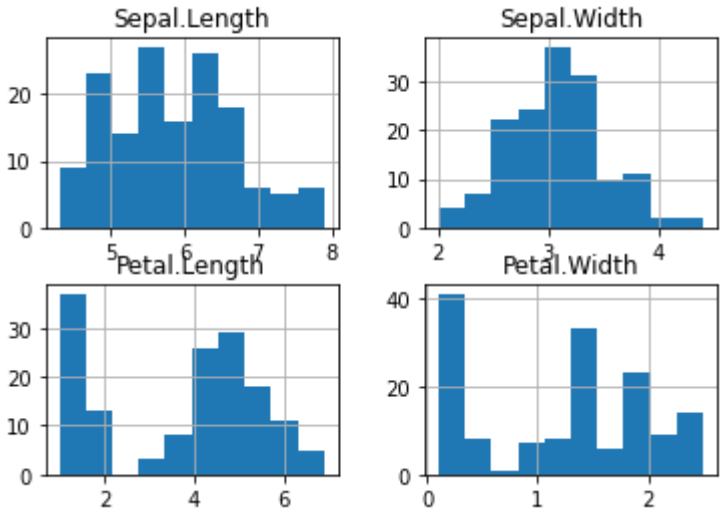


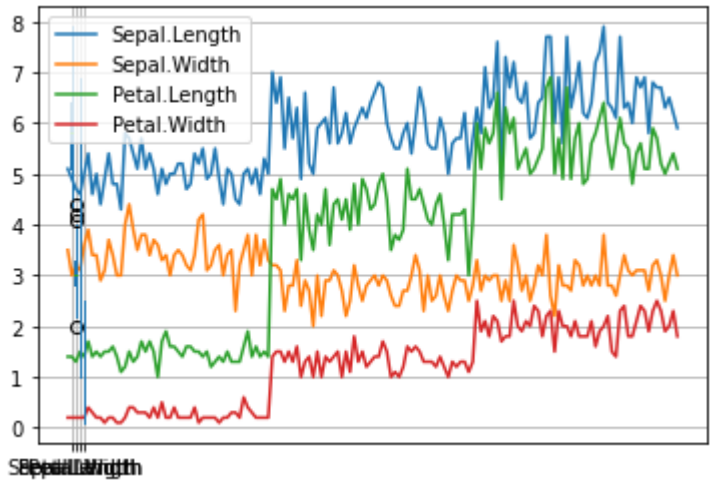
In [4]:

```
#Example 02
import pandas as pd
df = pd.read_csv('iris.csv')
df.hist()# Histogram
df.plot() # Line Graph
df.boxplot() # Box plot
```

Out[4]:

<AxesSubplot:>





In [6]:

```
from sklearn import datasets
import numpy as np
from sklearn import preprocessing
iris = datasets.load_iris()
X = iris.data[:, [2, 3]]
y = iris.target
std_scale = preprocessing.StandardScaler().fit(X)
X_std = std_scale.transform(X)
minmax_scale = preprocessing.MinMaxScaler().fit(X)
X_minmax = minmax_scale.transform(X)
```

In [20]:

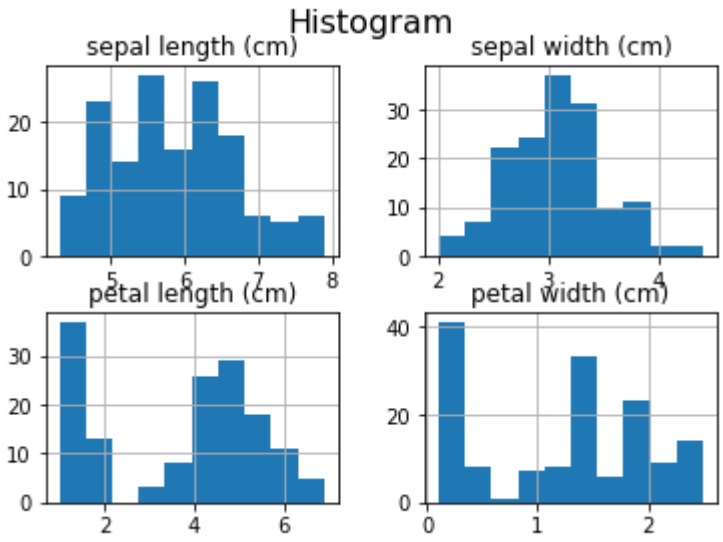
```
from sklearn import datasets #importing the database
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
iris = datasets.load_iris()
# Let's convert to dataframe
iris = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
columns= iris['feature_names'] + ['species'])
# replace the values with class labels
iris.species = np.where(iris.species == 0.0, 'setosa', np.where(iris.species==1.0,'versi
# let's remove spaces from column name
iris.columns = iris.columns.str.replace('','')
iris.describe()
#The columns 'species' is categorical, so lets check the frequency distribution for each
#category.
print (iris['species'].value_counts())
```

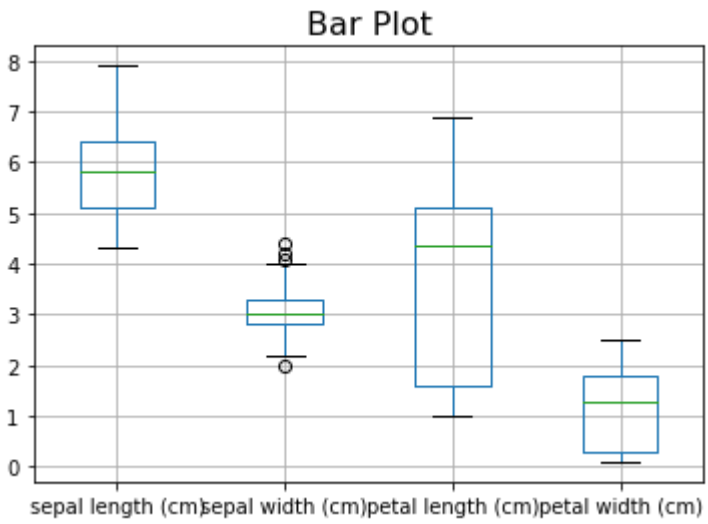
```
setosa        50
virginica     50
versicolor    50
Name: species, dtype: int64
```

In [21]:

```
iris.hist() # plot histogram
plt.suptitle("Histogram", fontsize=16) # use suptitle to add title to all
plt.show()

iris.boxplot() # plot boxplot
plt.title("Bar Plot", fontsize=16)
plt.show()
```





In [22]:

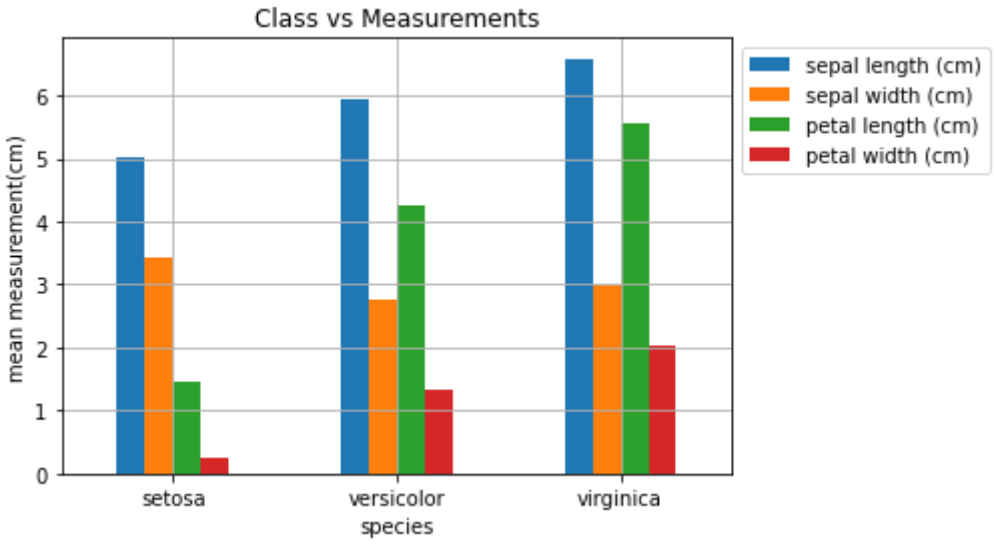
```
print(iris.groupby(by = "species").mean())
# plot for mean of each feature for each label class
iris.groupby(by = "species").mean().plot(kind="bar")
plt.title('Class vs Measurements')
plt.ylabel('mean measurement(cm)')
plt.xticks(rotation=0) # manage the xticks rotation
plt.grid(True)
# Use bbox_to_anchor option to place the legend outside plot area to be tidy
plt.legend(loc="upper left", bbox_to_anchor=(1,1))
```

```
            sepal length (cm)  sepal width (cm)  petal length (cm)  \
species
setosa                  5.006             3.428              1.462
versicolor              5.936             2.770              4.260
virginica               6.588             2.974              5.552

            petal width (cm)
species
setosa                 0.246
versicolor             1.326
virginica              2.026
```

Out[22]:

```
<matplotlib.legend.Legend at 0x2564f0f6160>
```

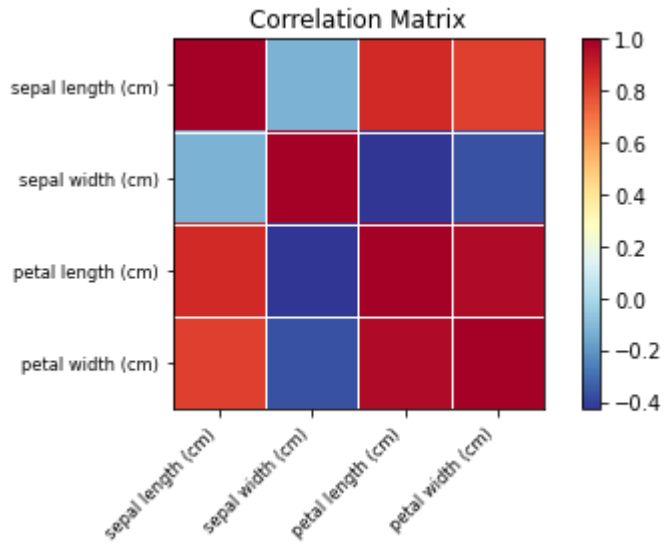


In [23]:

```
# create correlation matrix
corr = iris.corr()
print(corr)
import statsmodels.api as sm
sm.graphics.plot_corr(corr, xnames=list(corr.columns))
plt.show()
```

```
                   sepal length (cm)  sepal width (cm)  petal length (cm)
\
sepal length (cm)           1.000000         -0.117570           0.871754
sepal width (cm)           -0.117570          1.000000          -0.428440
petal length (cm)           0.871754         -0.428440           1.000000
petal width (cm)            0.817941         -0.366126           0.962865

                   petal width (cm)
sepal length (cm)          0.817941
sepal width (cm)          -0.366126
petal length (cm)          0.962865
petal width (cm)           1.000000
```

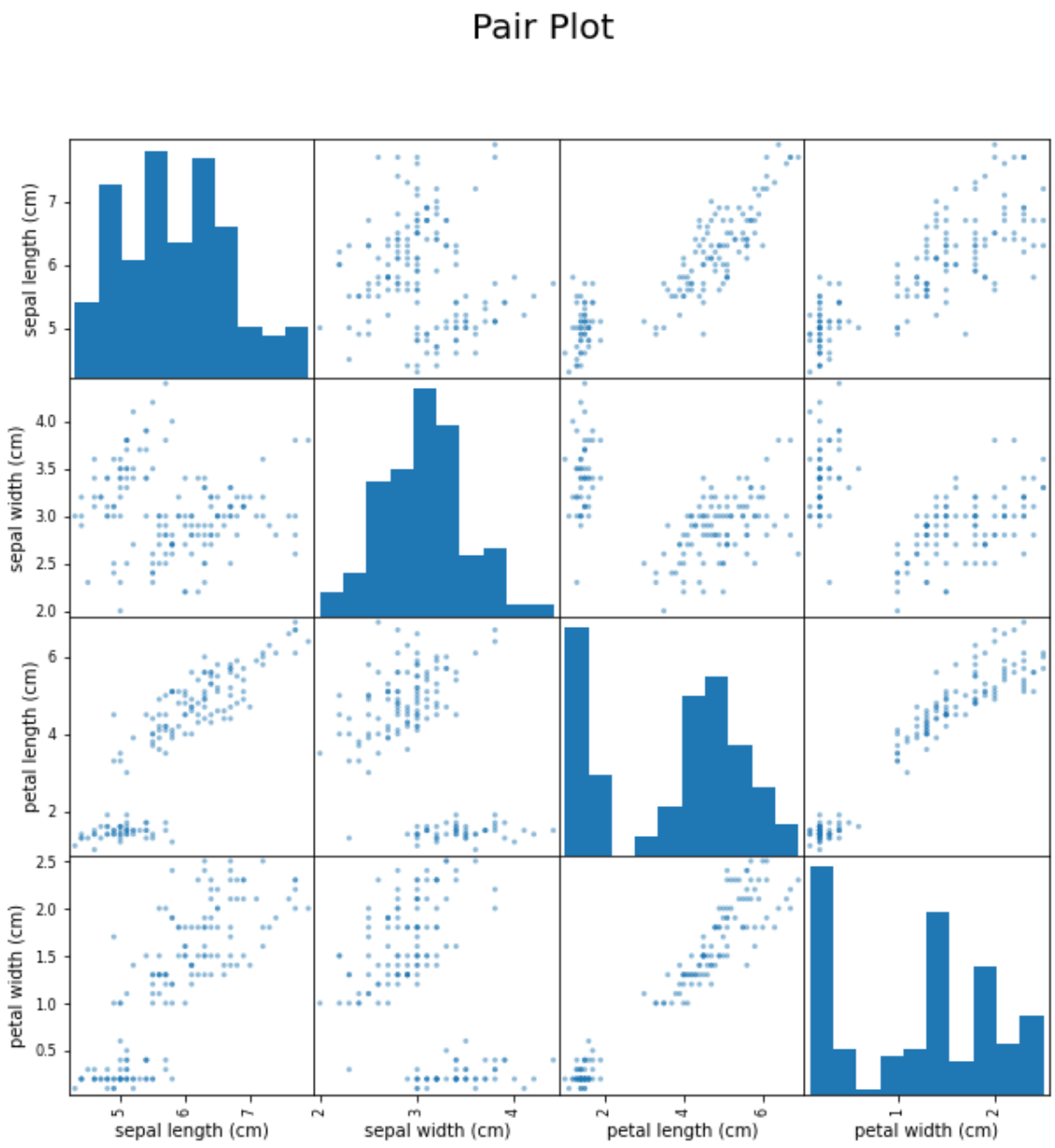


In [37]:

```
from pandas.plotting import scatter_matrix
scatter_matrix(iris, figsize=(10, 10))
# use suptitle to add title to all sublots
plt.suptitle("Pair Plot", fontsize=20)
```

Out[37]:

```
Text(0.5, 0.98, 'Pair Plot')
```



In [ ]: