# KMeans Clustering

## Iris data from Scikit learn package

### Data loaded

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
from sklearn.cluster import KMeans
from sklearn.preprocessing import scale
import seaborn as sns
```

In [2]:
```
from sklearn import datasets as dat
iris=dat.load_iris()
```

In [3]:
```
df=pd.DataFrame(iris.data,columns=iris.feature_names)
df1=df[['sepal length (cm)','petal length (cm)']]
df1.head()
```

Out[3]:

| | sepal length (cm) | petal length (cm) |
|---|---|---|
| **0** | 5.1 | 1.4 |
| **1** | 4.9 | 1.4 |
| **2** | 4.7 | 1.3 |
| **3** | 4.6 | 1.5 |
| **4** | 5.0 | 1.4 |

## Preprocessing

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
dtypes: float64(4)
memory usage: 4.8 KB
```

In [5]:
```
df.describe()
```

Out[5]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) |
|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.057333 | 3.758000 | 1.199333 |
| **std** | 0.828066 | 0.435866 | 1.765298 | 0.762238 |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
dtype: int64
```

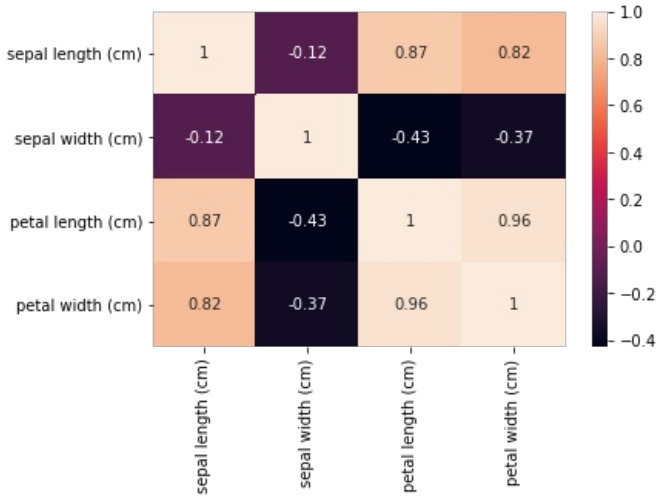
In [7]:
```
df.corr()
```

Out[7]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) |
|---|---|---|---|---|
| **sepal length (cm)** | 1.000000 | -0.117570 | 0.871754 | 0.817941 |
| **sepal width (cm)** | -0.117570 | 1.000000 | -0.428440 | -0.366126 |
| **petal length (cm)** | 0.871754 | -0.428440 | 1.000000 | 0.962865 |
| **petal width (cm)** | 0.817941 | -0.366126 | 0.962865 | 1.000000 |

In [8]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[8]:
```
<AxesSubplot:>
```



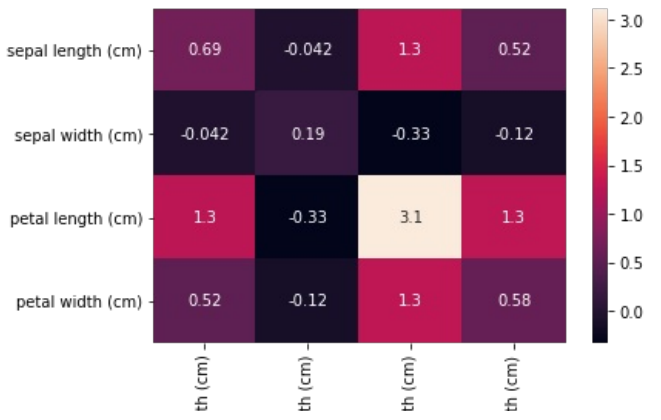

In [9]:
```
df.cov()
```

Out[9]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) |
|---|---|---|---|---|
| **sepal length (cm)** | 0.685694 | -0.042434 | 1.274315 | 0.516271 |
| **sepal width (cm)** | -0.042434 | 0.189979 | -0.329656 | -0.121639 |
| **petal length (cm)** | 1.274315 | -0.329656 | 3.116278 | 1.295609 |
| **petal width (cm)** | 0.516271 | -0.121639 | 1.295609 | 0.581006 |

In [10]:
```
sns.heatmap(df.cov(),annot=True)
```

Out[10]:
```
<AxesSubplot:>
```

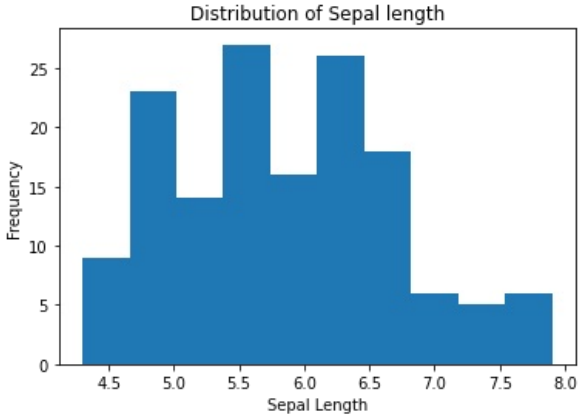


sepal leng sepal widt petal leng petal widt

## Exploratory Data Analysis

In [11]:
```
df['sepal length (cm)'].plot.hist()
plt.xlabel('Sepal Length')
plt.title('Distribution of Sepal length')
```

Out[11]: Text(0.5, 1.0, 'Distribution of Sepal length')

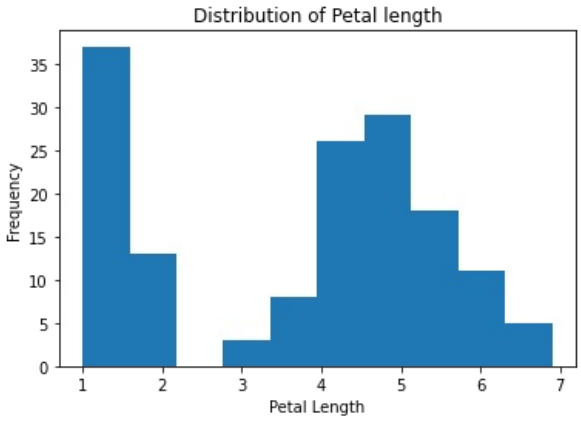



In [12]:
```
df.count()
```

Out[12]:
```
sepal length (cm)    150
sepal width (cm)     150
petal length (cm)    150
petal width (cm)     150
dtype: int64
```

In [13]:
```
df['petal length (cm)'].plot.hist()
plt.xlabel('Petal Length')
plt.title('Distribution of Petal length')
```

Out[13]: Text(0.5, 1.0, 'Distribution of Petal length')




In [14]:
```
x=df1['sepal length (cm)']
y=df1['petal length (cm)']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.23,random_state=2)
```

In [15]:
```
df.head()
```

Out[15]:

| | 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 |

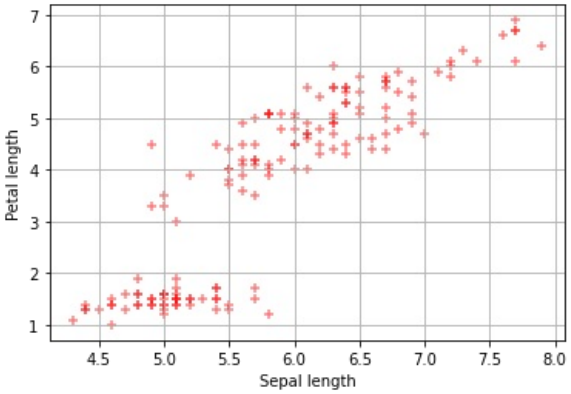
In [16]:

```
fig=plt.figure(figsize=(10,5))
x=df1['sepal length (cm)']
y=df1['petal length (cm)']

fig,ax=plt.subplots()

ax.scatter(x,y,marker='+',c='red',alpha=0.5)
plt.grid()
plt.xlabel('Sepal length')
plt.ylabel('Petal length')
```

Out[16]:

```
Text(0, 0.5, 'Petal length')
```

```
<Figure size 720x360 with 0 Axes>
```




In [17]:

```
from sklearn.cluster import KMeans

kmeans=KMeans(n_clusters=3)
kmeans.fit(df1)
```

Out[17]:

```
KMeans(n_clusters=3)
```

In [18]:

```
labels=kmeans.predict(df1)
centroids=kmeans.cluster_centers_
```

In [19]:

```
centroids
```

Out[19]:

```
array([[5.87413793, 4.39310345],
       [5.00784314, 1.49215686],
       [6.83902439, 5.67804878]])
```

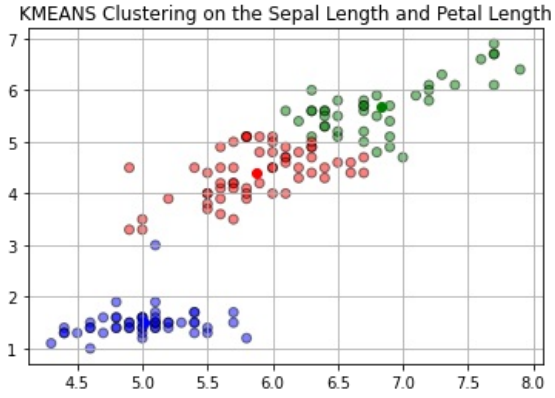
## Evaluation

In [20]:

```
fig=plt.figure(figsize=(10,7))

colmap={1:'r',2:'b',3:'g'}

colors=map(lambda x:colmap[x+1],labels)
colors1=list(colors)

fig,ax=plt.subplots()
```

```
ax.scatter(x,y,color=colors1,alpha=0.5,edgecolor='k')

for idx,centroid in enumerate(centroids):
    plt.scatter(*centroid, color=colmap[idx+1])

plt.title('KMEANS Clustering on the Sepal Length and Petal Length')
plt.grid()
```

```
<Figure size 720x504 with 0 Axes>
```




Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js