K- Means Clustering

To Explore Unsupervised Machine Learning

Loading the Libraries

In [121]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import datasets
from sklearn.cluster import KMeans
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
import warnings
warnings.filterwarnings("ignore")
```

Display the Dataset

In [122]:
```
iris=pd.read_csv("C:Documents\Iris_dataset.csv")
iris
```

Out[122]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spe
0	1	5.1	3.5	1.4	0.2	se
1	2	4.9	3.0	1.4	0.2	se
2	3	4.7	3.2	1.3	0.2	se
3	4	4.6	3.1	1.5	0.2	se
4	5	5.0	3.6	1.4	0.2	se
...	...	...	...	...	...	

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Spe
145	146	6.7	3.0	5.2	2.3	virg
146	147	6.3	2.5	5.0	1.9	virg
147	148	6.5	3.0	5.2	2.0	virg
148	149	6.2	3.4	5.4	2.3	virg
149	150	5.9	3.0	5.1	1.8	virg

150 rows × 6 columns

Removing first column

In [123]:
```
iris=iris.drop('Id',axis=1)
iris.head()
```

Out[123]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Analysing the Dataset

In [124]:
```
iris.shape # dimension of data contains 150 rows and 5 cols
```

Out[124]:
```
(150, 5)
```

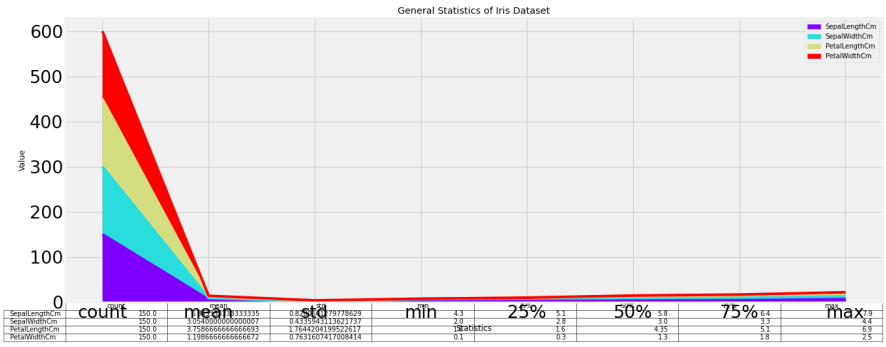
In [125]:
```
iris.isnull().sum()# missing values
```

Out[125]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
```

```
Species          0
dtype: int64
```

In [126]:
```
# displaying the descriptive statistics of dataset
iris.describe().plot(kind = "area",fontsize=27, figsize = (20,8), t
able = True,colormap="rainbow")
plt.xlabel('Statistics',)
plt.ylabel('Value')
plt.title("General Statistics of Iris Dataset")
```

Out[126]:
```
Text(0.5, 1.0, 'General Statistics of Iris Dataset')
```

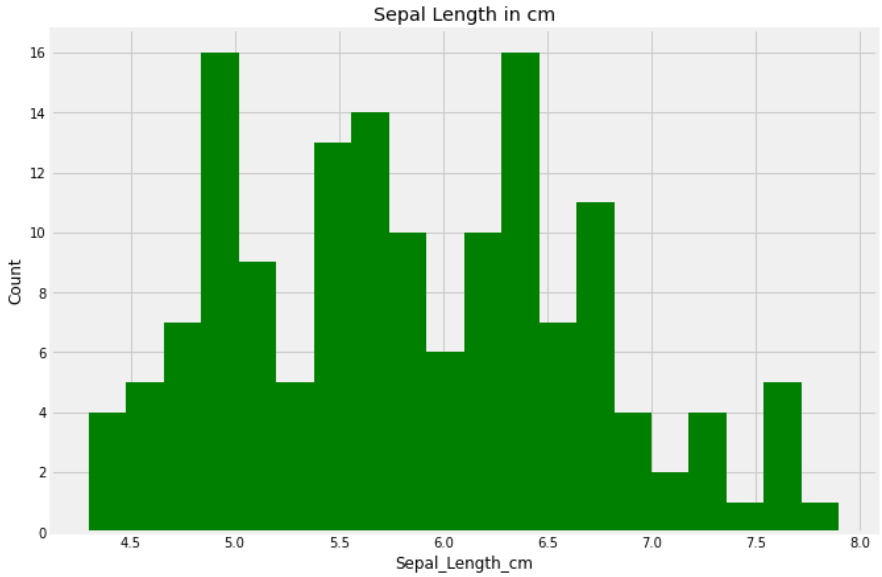


In [127]:
```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   SepalLengthCm   150 non-null    float64
 1   SepalWidthCm    150 non-null    float64
 2   PetalLengthCm   150 non-null    float64
 3   PetalWidthCm    150 non-null    float64
 4   Species         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

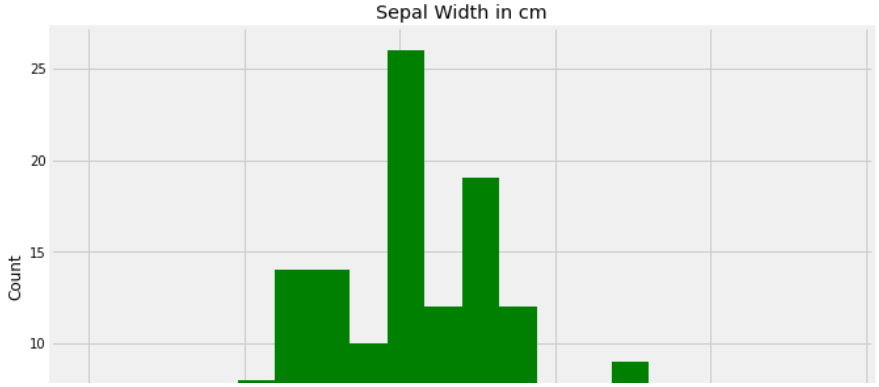
In [128]:
```
plt.figure(figsize = (10, 7))
x = iris["SepalLengthCm"]

plt.hist(x, bins = 20, color = "green")
plt.title("Sepal Length in cm")
plt.xlabel("Sepal_Length_cm")
plt.ylabel("Count")
```

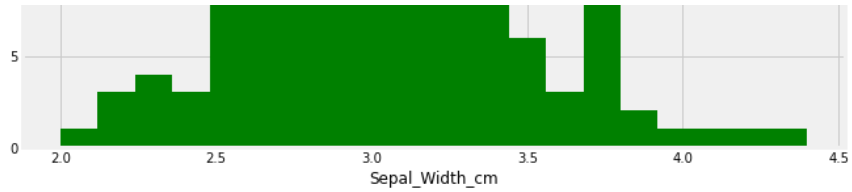
Out[128]:
```
Text(0, 0.5, 'Count')
```



In [129]:

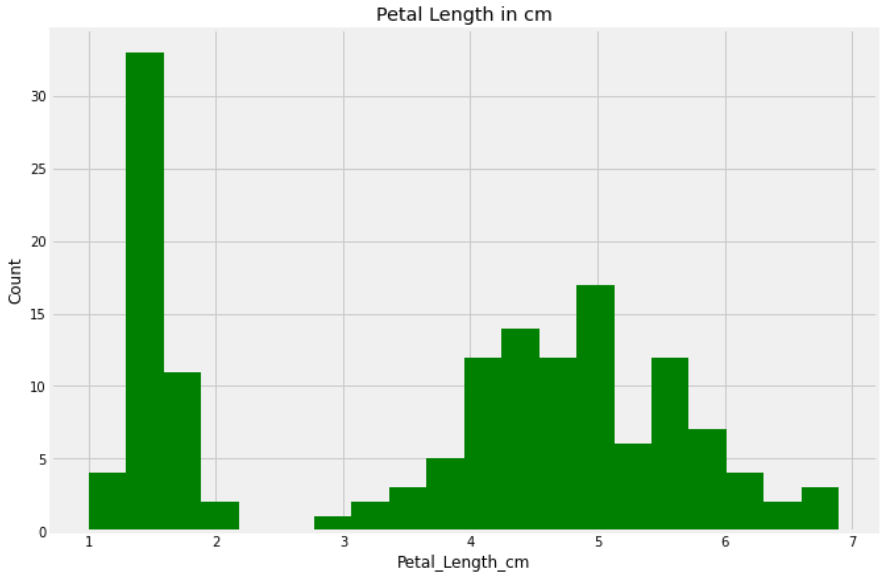
```
plt.figure(figsize = (10, 7))
x = iris.SepalWidthCm

plt.hist(x, bins = 20, color = "green")
plt.title("Sepal Width in cm")
plt.xlabel("Sepal_Width_cm")
plt.ylabel("Count")

plt.show()
```





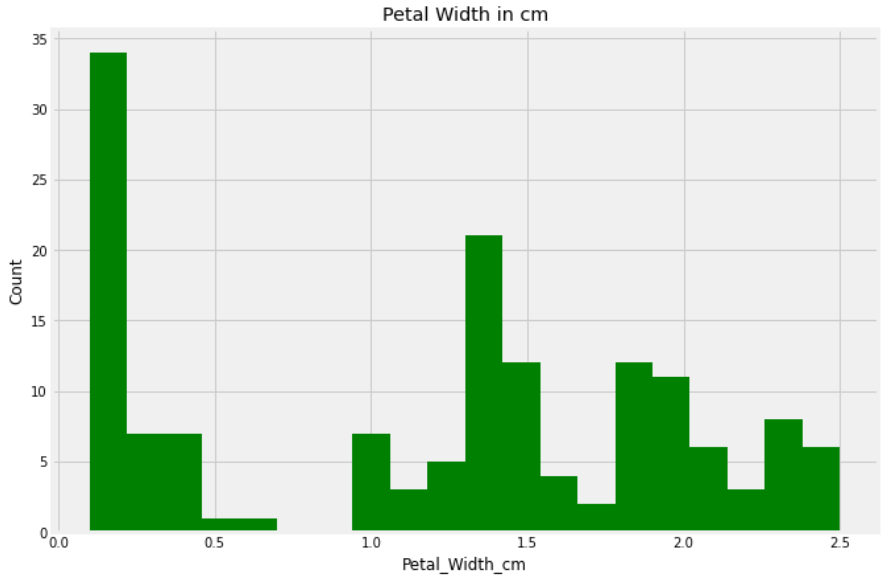
In [130]:

```
plt.figure(figsize = (10, 7))
x = iris.PetalLengthCm

plt.hist(x, bins = 20, color = "green")
plt.title("Petal Length in cm")
plt.xlabel("Petal_Length_cm")
plt.ylabel("Count")

plt.show()
```



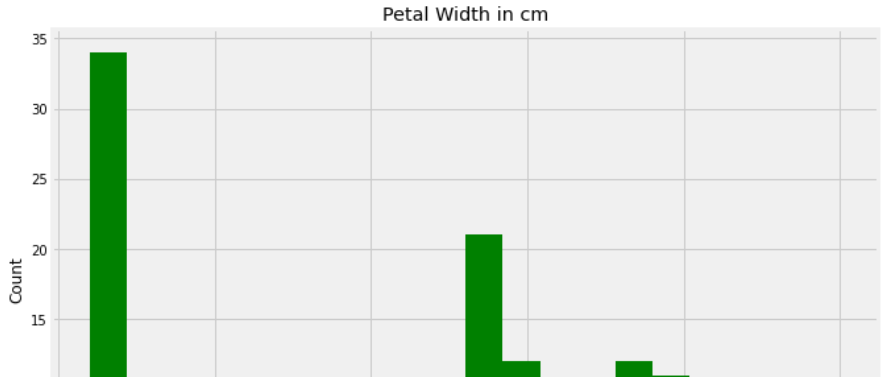
In [131]:

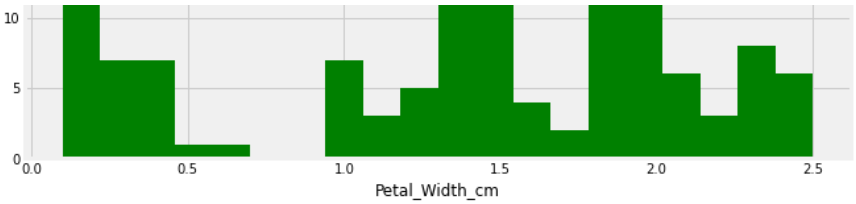
```
plt.figure(figsize = (10, 7))
x = iris.PetalWidthCm

plt.hist(x, bins = 20, color = "green")
plt.title("Petal Width in cm")
plt.xlabel("Petal_Width_cm")
plt.ylabel("Count")

plt.show()
```



In [132]:

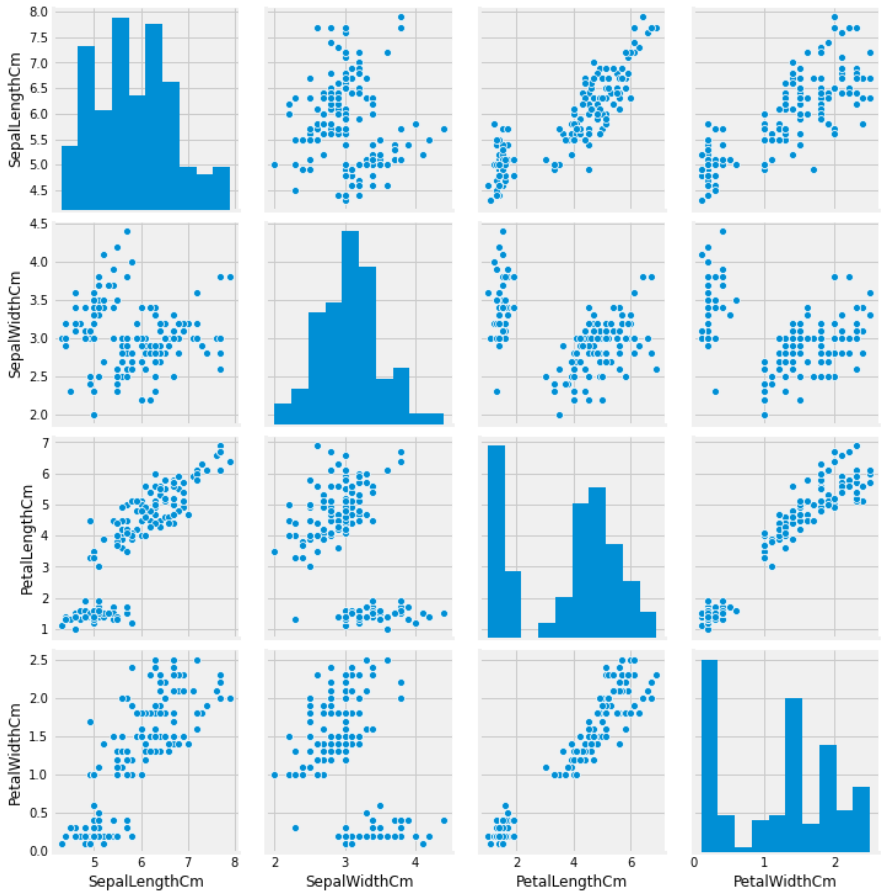
```
plt.figure(figsize = (10, 7))
x = data.PetalWidthCm

plt.hist(x, bins = 20, color = "green")
plt.title("Petal Width in cm")
plt.xlabel("Petal_Width_cm")
plt.ylabel("Count")

plt.show()
```





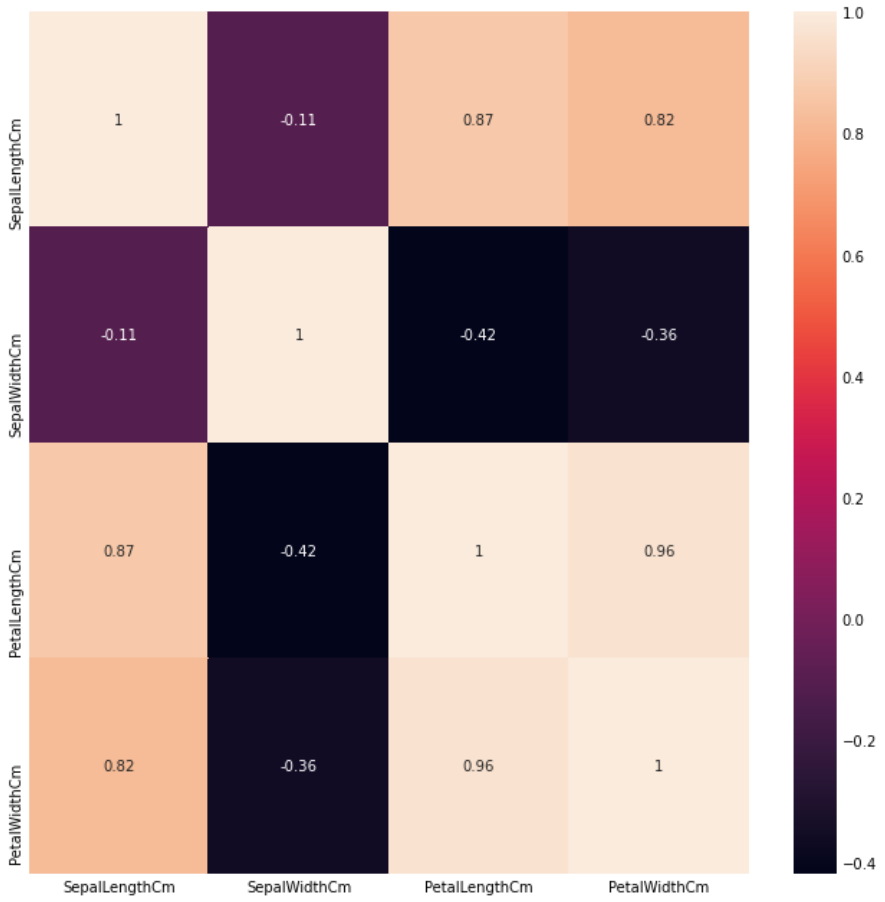
In [133]:
```
sns.pairplot(iris)
```

Out[133]: <seaborn.axisgrid.PairGrid at 0x25c2cf45910>



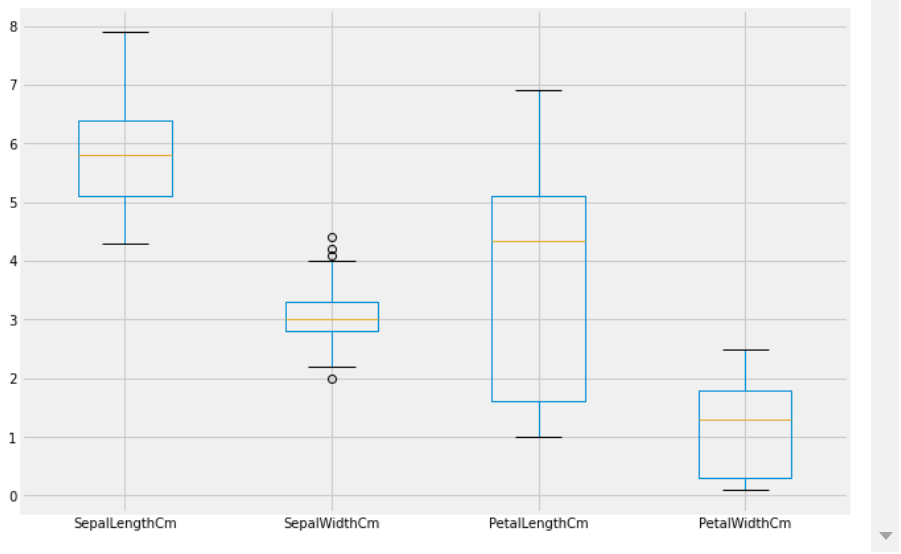
In [134]:
```
plt.figure(figsize=(10,11))
sns.heatmap(iris.corr(),annot=True)
plt.plot()
```

Out[134]: []



In [135]:
```
plt.figure(figsize = (10, 7))
iris.boxplot()
```

Out[135]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25c2d1f6fa0>
```



In [139]:
```
#Iris Dataset
iris = datasets.load_iris()
X = iris.data
```

In [140]:
```
#KMeans
km = KMeans(n_clusters=3)
km.fit(X)
km.predict(X)
labels = km.labels_
```

In [120]:
```
plt.scatter(X[:,0], X[:,1], c=y, cmap='gist_rainbow')
plt.xlabel('Spea1 Length', fontsize=18)
plt.ylabel('Sepal Width', fontsize=18)
```

Out[120]:
```
Text(0, 0.5, 'Sepal Width')
```