

IRIS DATASET VISUALIZATIONS(SEABORN,MATPLOTLIB)

I have created this kernel for beginners who want to learn how to plot graphs with seaborn.this kernel is still a work in progress .I will be updating it durther when i found some time .if you find my work useful please to vote by clicking at the top of the page.thanks for viewing.

```
In [91]: import numpy as np
import pandas as pd
```

Importing pandas and seaborn module

```
In [94]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#plt.style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore')
```

Importing iris data set

```
In [97]: iris=pd.read_csv(r"C:\Users\krishna\OneDrive\Desktop\work\28th -Seaborn movie an
```

```
In [99]: iris
```

Out[99]:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|------------|-----------|----------------------|---------------------|----------------------|---------------------|----------------|
| 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| ... | ... | ... | ... | ... | ... | ... |
| 145 | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| 146 | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| 147 | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| 148 | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| 149 | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

150 rows × 6 columns

In [101... `iris.head()`

Out[101...]

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|----------|-----------|----------------------|---------------------|----------------------|---------------------|----------------|
| 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

In [103... `iris.drop('Id',axis=1,inplace=True)`In [105... `iris.head()`

Out[105...

| | 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 |

In [107... `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
```

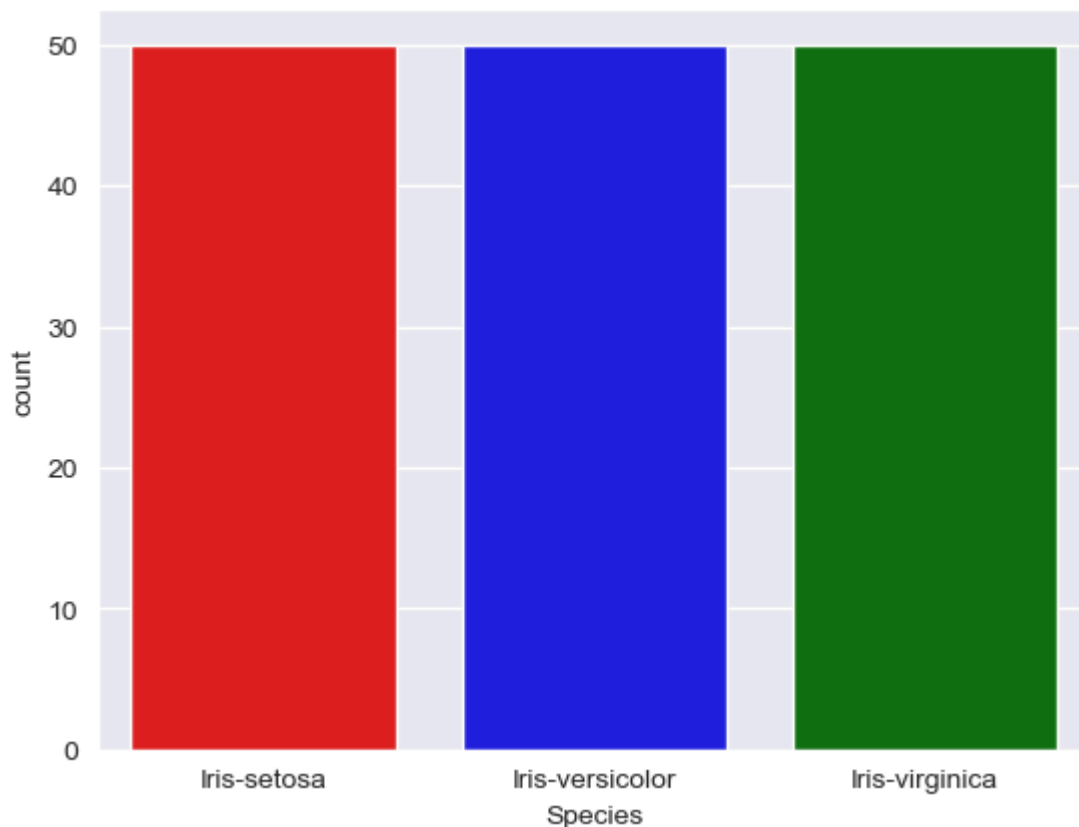
In [109... `iris['Species'].value_counts()`

Out[109... Species
Iris-setosa 50
Iris-versicolor 50
Iris-virginica 50
Name: count, dtype: int64

This data set has three varieties of iris plant.

2.bar plot : here the frequency of the observation is plotted.in this case we are plotting the frequency of the three species in the iris dataset.

In [113... `sns.countplot(x='Species', data=iris,palette=["red", "blue", "green"])`
`plt.show()`



We can send that there are 50 samples each of all the iris species in the data set.

4. joint plot: jointplot is seaborn library specific and can be used to quickly visualize and analyze the relationship between two variables and describe their individual distributions on the same plot.

```
In [117...] iris.head()
```

```
Out[117...]
   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
```

```
In [119...] fig=sns.jointplot(x='SepalLengthCm', y='SepalWidthCm',data= iris)
```

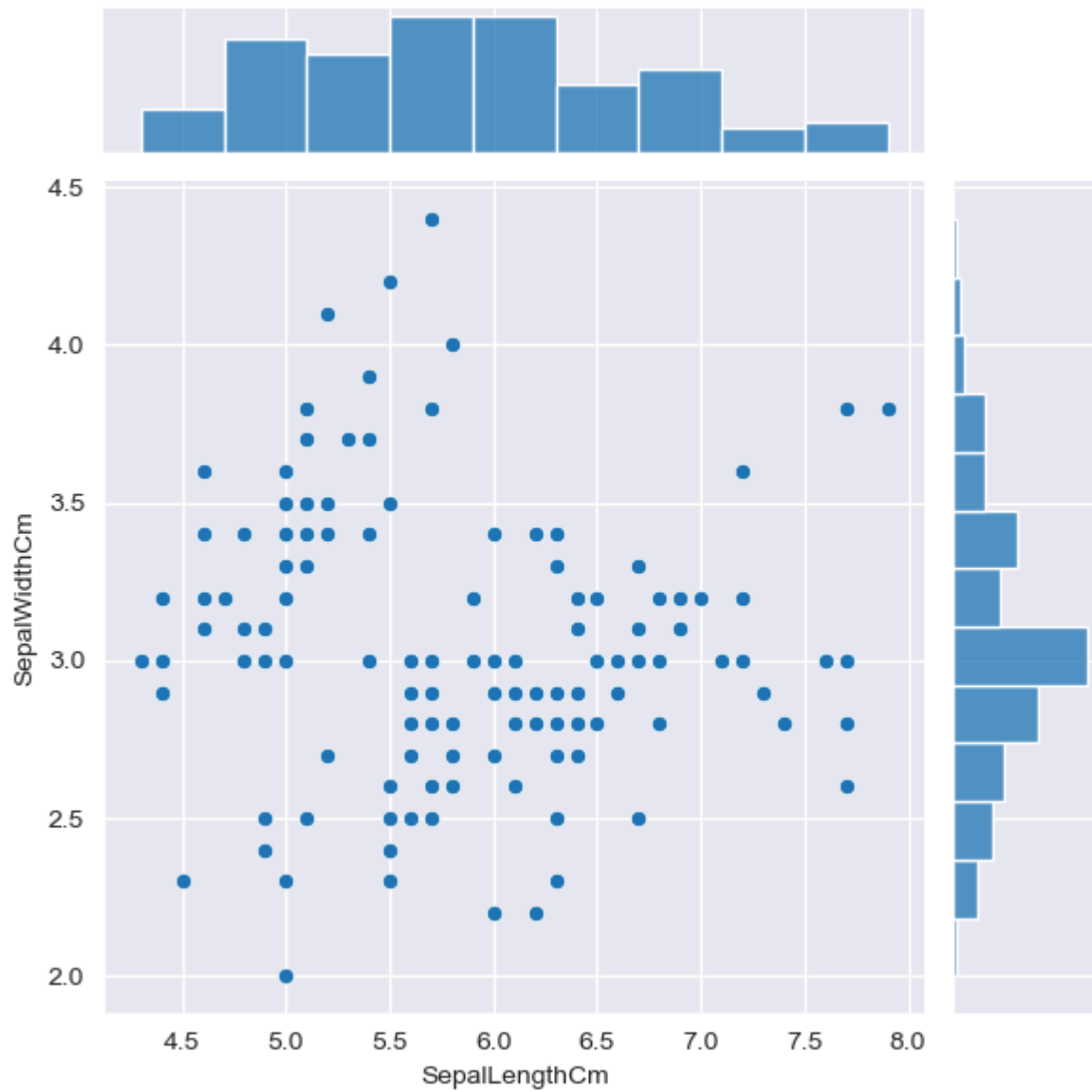
```
In [120...] fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind = 'hex',data =iris)
```

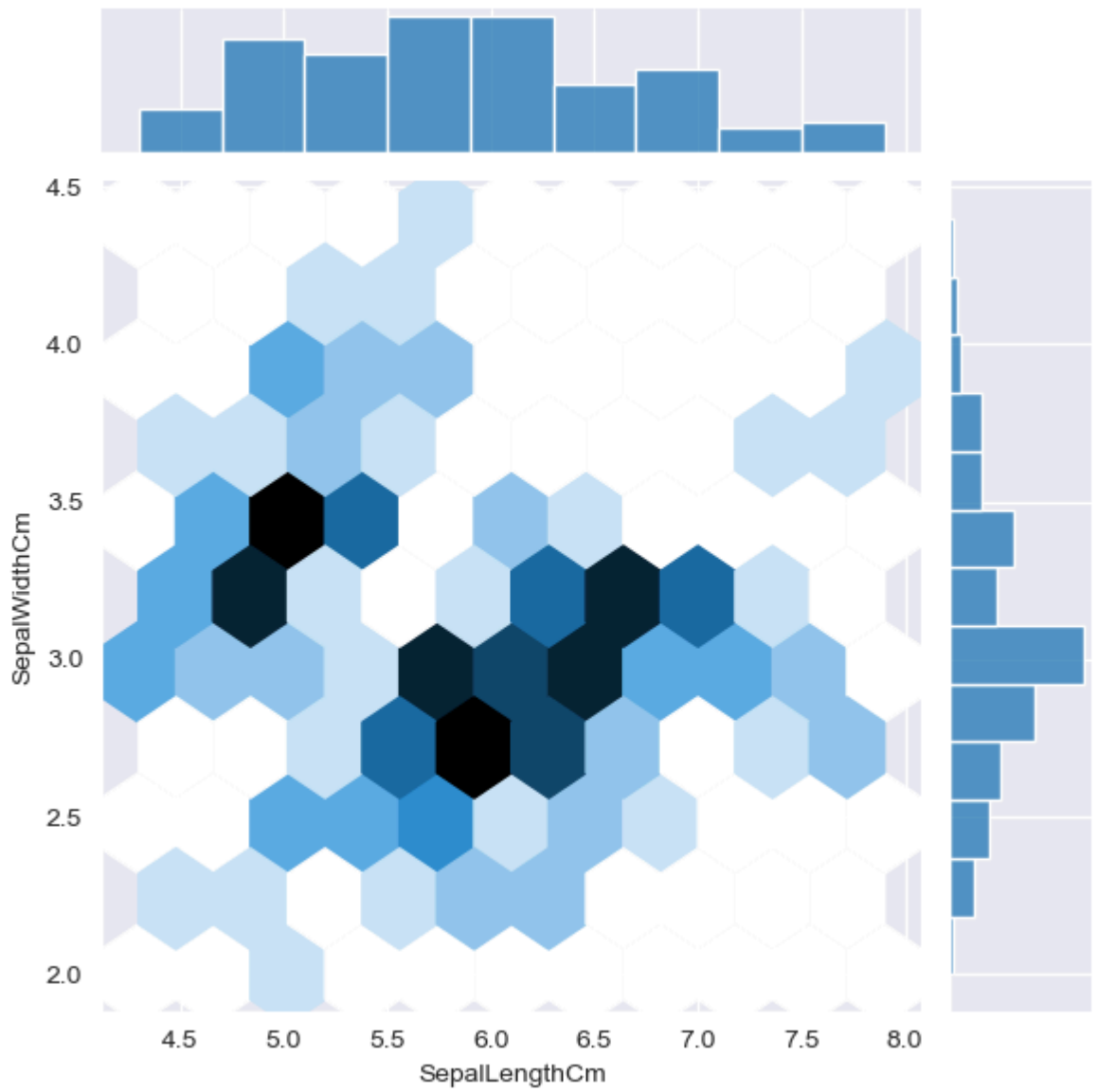
5. FacetGrid plot

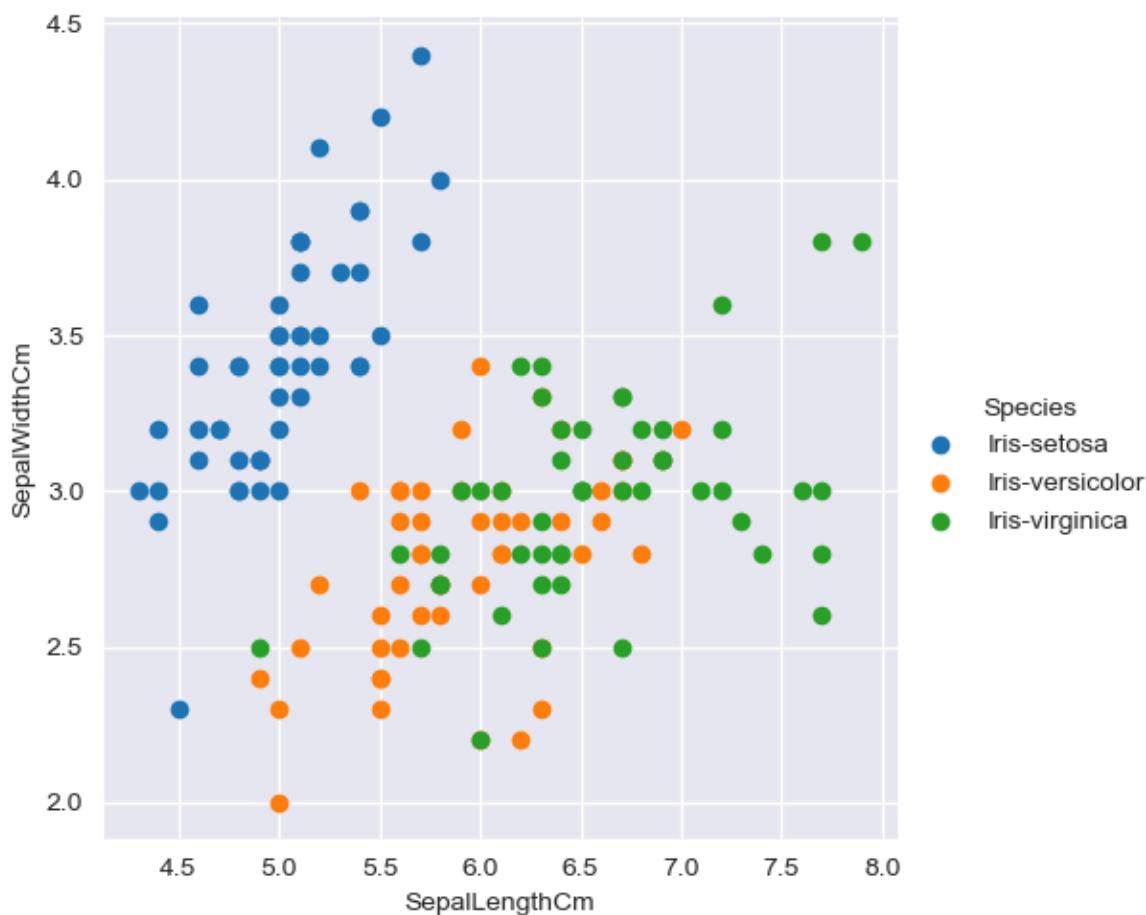
```
In [123...] import matplotlib.pyplot as plt
%matplotlib inline

sns.FacetGrid(iris, hue='Species', height = 5)\
.map(plt.scatter,'SepalLengthCm','SepalWidthCm')\
```

```
.add_legend()  
plt.show()
```







6.Boxplot or Whisker plot box plot was first introduced in year 1969 by mathematician john tukey.box plot give a statical summary of the features being plotted top line represent the max value,top edge of box is third quartile middle edge represents the median,bottom edge represents the first quartile value.The bottom most line respresent the minimum value of the feature.The height of the box is called as Interquartile range.The black dots on the plot represent the outlier values in the data.

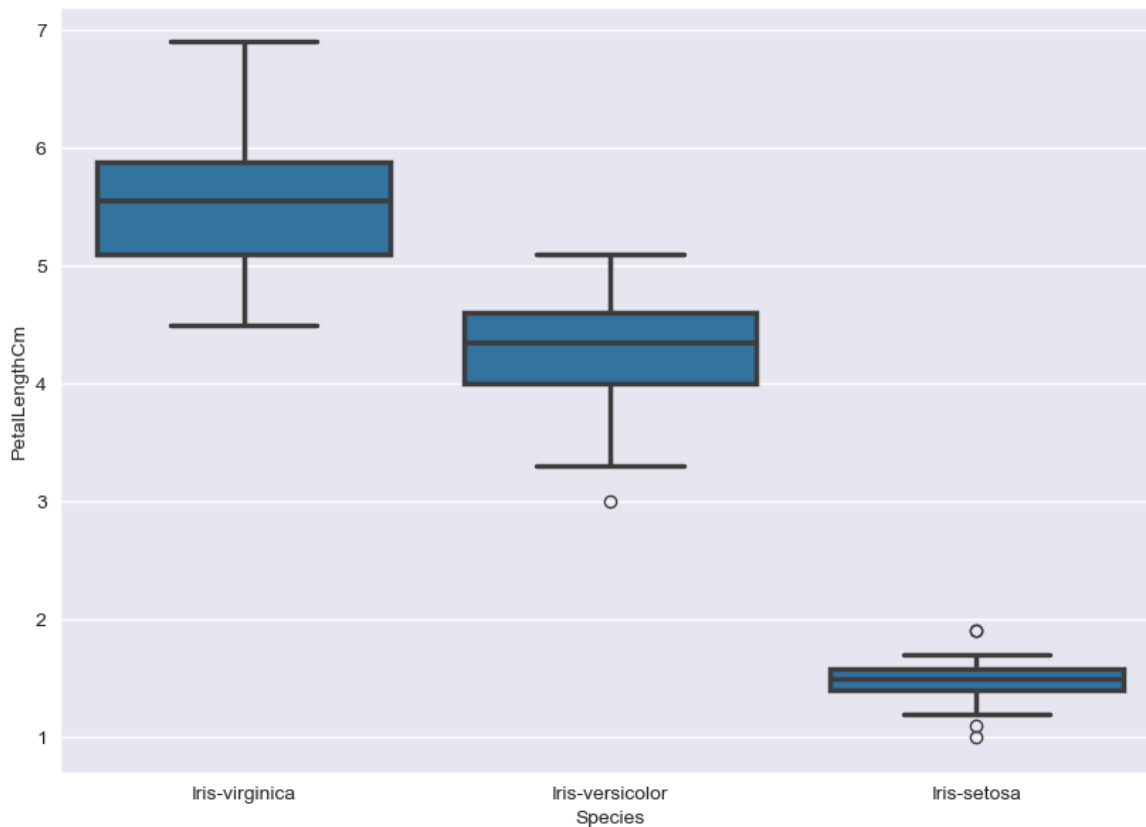
In [126... `iris.head()`

Out[126...

| | 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 |

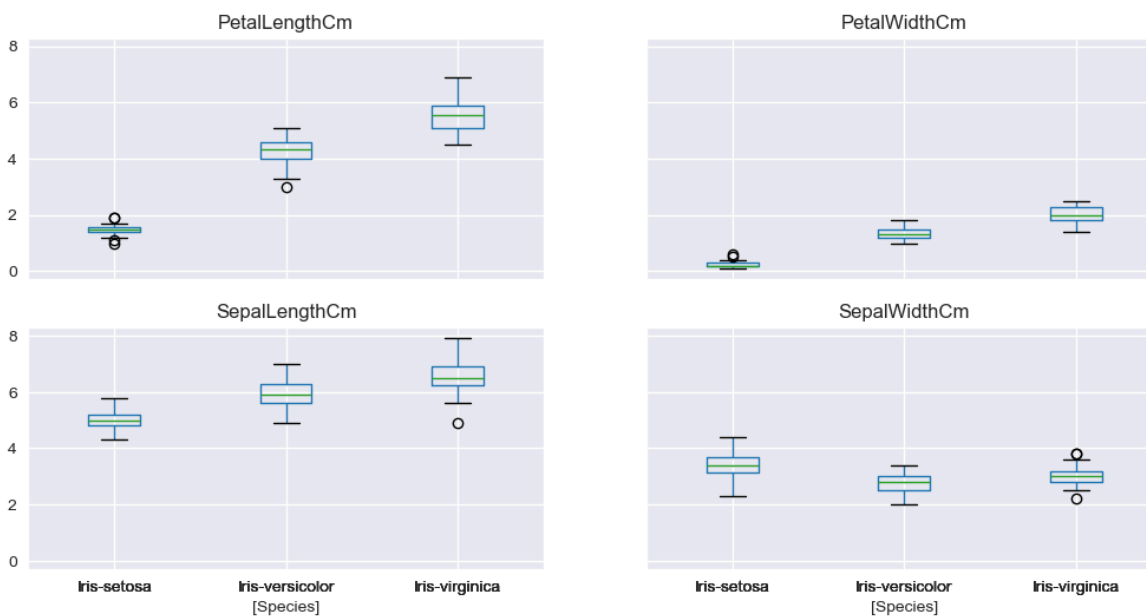
In [127...

```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species', y= 'PetalLengthCm', data = iris,order=['Iris-virginica','Iris-versicolor','Iris-setosa'])
plt.show()
```



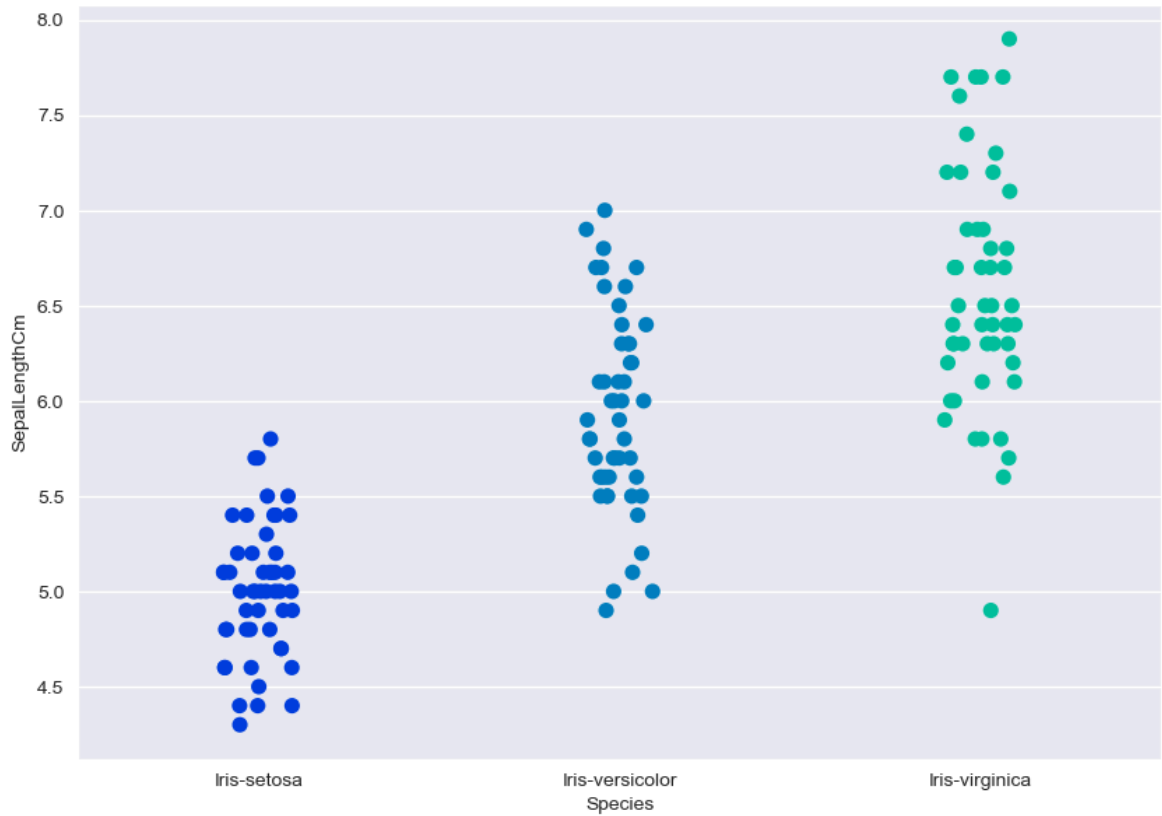
```
In [130... #iris.drop("Id", axis=1).boxplot(by="Species", figsize=(12, 6))
iris.boxplot(by="Species", figsize=(12, 6))
plt.show()
```

Boxplot grouped by Species



7.Strip plot

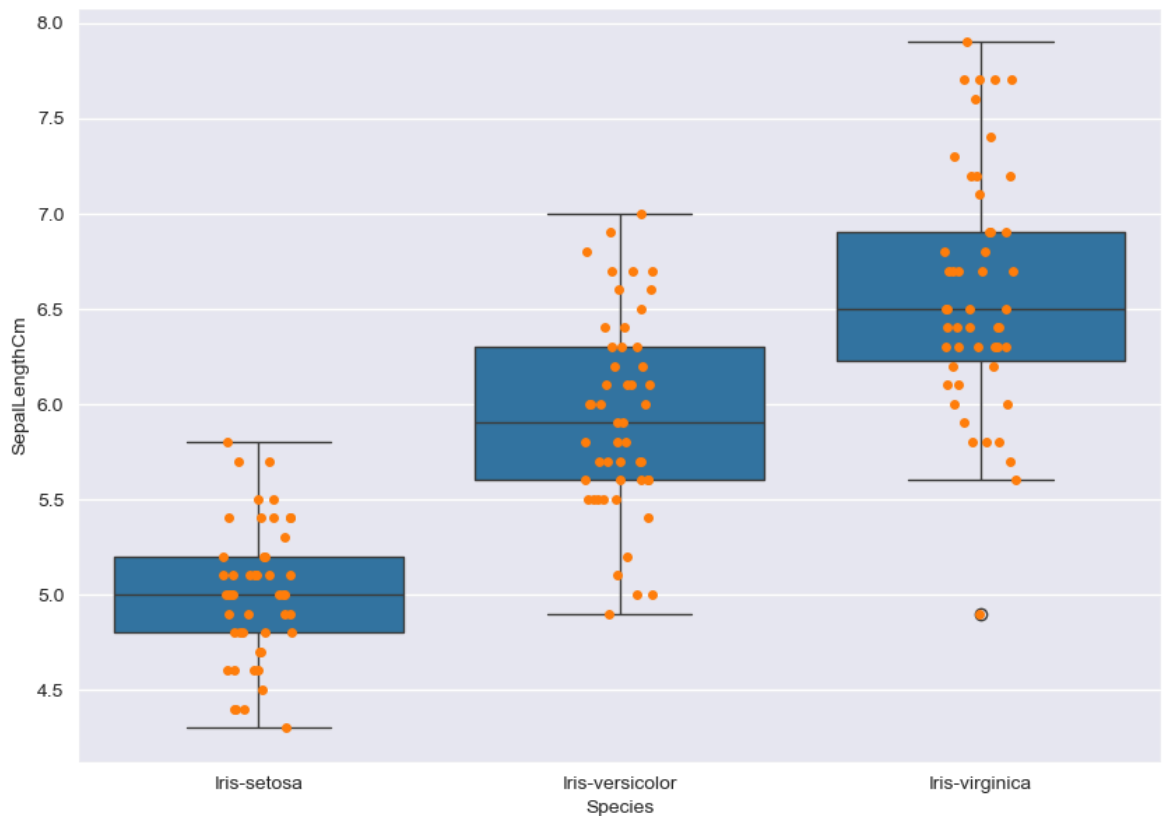
```
In [133... fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data = iris, jitter= True, edgec
plt.show()
```

8. Combining Box and Strip Plots

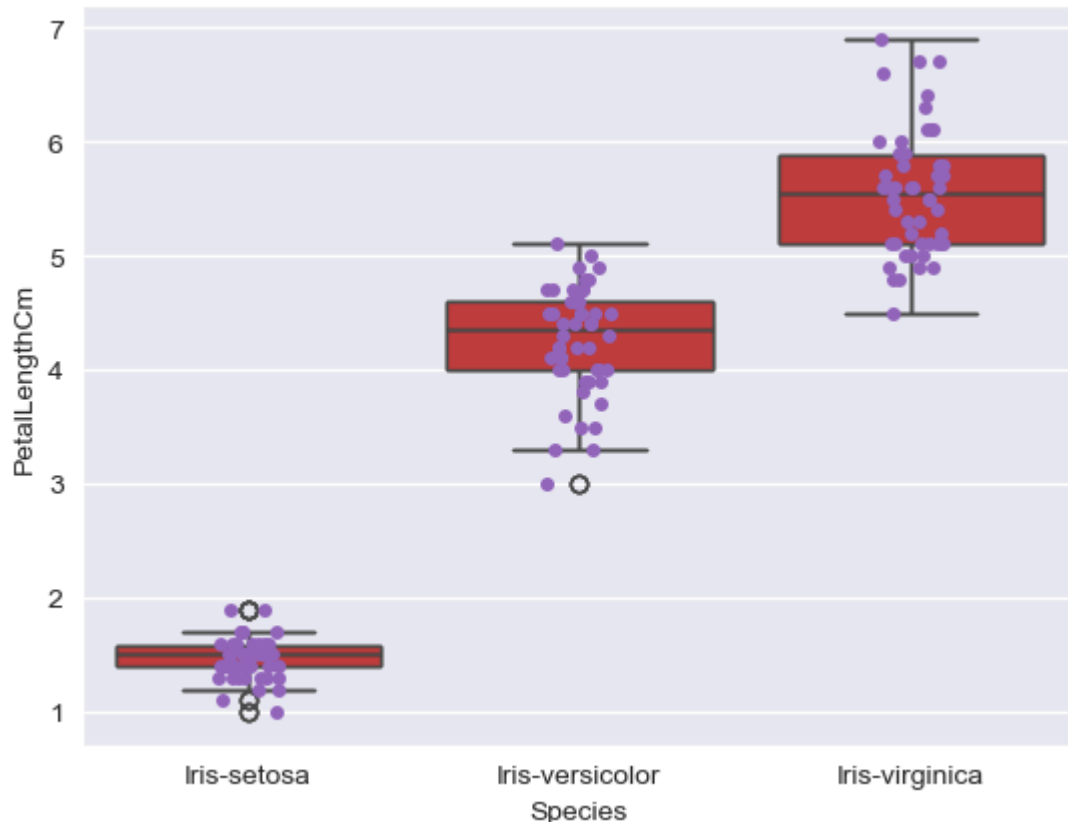
In [135...

```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data =iris)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor=
plt.show()
```



```
In [188... sx=sns.boxplot(x="Species",y="PetalLengthCm", data= iris)
sx=sns.stripplot(x="Species",y="PetalLengthCm", data= iris,jitter =True,edgecolor=

box_colors = ["green", "red", "yellow"]
if len(sx.patches) >= 3: # Ensure at Least 3 boxes exist before modifying
    for i, box in enumerate(sx.patches[:3]): # Modify first 3 boxes
        box.set_facecolor(box_colors[i])
        box.set_edgecolor("black")
plt.show()
```



9.Violin plot it is used to visualize the distribution of data and its probability distribution.this chart is a combination of a Box plot and a Density plot that is rotated and placed on each side ,to show the distribution shape of the data . the thick bar in the centre represents the interquartile range ,the thin black line extended from it represents the 95% confidence intervals, and the white dots is the median.box plots are limited in their display of the data as their visual simplicity tends to hide significant details about how values in the data are distributed

```
In [ ]: fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',data=iris)
plt.show()
```

```
In [ ]: plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=iris)
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=iris)
```

```
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris)
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalWidthCm',data=iris)
plt.show()
```

10. Pair plot: A "pairs plot" is also known as a scatterplot, in which one variable in the same data row is matched with another variables value like this : pairs plots are just elaborations on this showing all variables paired with all the other variables.

```
In [ ]: sns.pairplot(data = iris,kind='scatter')
plt.show()
```

```
In [ ]: sns.pairplot(iris,hue ='Species');
plt.show()
```

11.Heat map Heat map is used to find out the correlation between different features have high correlation this helps us to select the parameters for machine learning.

```
In [141... fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.heatmap(iris.corr(),annot=True,cmap='cubehelix',linewidths=1,linecolor='
plt.show()
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[141], line 3
      1 fig=plt.gcf()
      2 fig.set_size_inches(10,7)
----> 3 fig=sns.heatmap(iris.corr(),annot=True,cmap='cubehelix',linewidths=1,line
color='k',square=True,mask=False, vmin=-1, vmax=1,cbar_kws={"orientation": "verti
cal"},cbar=True)
      4 plt.show()

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:11049, in DataFrame.corr
(self, method, min_periods, numeric_only)
    11047 cols = data.columns
    11048 idx = cols.copy()
> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
    11051 if method == "pearson":
    11052     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:1993, in DataFrame.to_num
py(self, dtype, copy, na_value)
    1991 if dtype is not None:
    1992     dtype = np.dtype(dtype)
-> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
    1994 if result.dtype is not dtype:
    1995     result = np.asarray(result, dtype=dtype)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1694, in Blo
ckManager.as_array(self, dtype, copy, na_value)
    1692     arr.flags.writeable = False
    1693 else:
-> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
    1695     # The underlying data was copied within _interleave, so no need
    1696     # to further copy if copy=True or setting na_value
    1698 if na_value is lib.no_default:

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1753, in Blo
ckManager._interleave(self, dtype, na_value)
    1751     else:
    1752         arr = blk.get_values(dtype)
-> 1753     result[rl.indexer] = arr
    1754     itemmask[rl.indexer] = 1
    1756 if not itemmask.all():

ValueError: could not convert string to float: 'Iris-setosa'

```

12.Distribution plot: the distribution plot is suitable for comparing range and distribution for groups of numerical data .data is plotted as value points along an axis you can choose to display only the value points to see the distribution of values,a bounding box to see the range of values ,or a combination of both as shown here.the distribution plot is not relevant for detailed analysis of the data as it delas with a summory of the data distribution

```

In [ ]: iris.hist(edgecolor='black',linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()

```

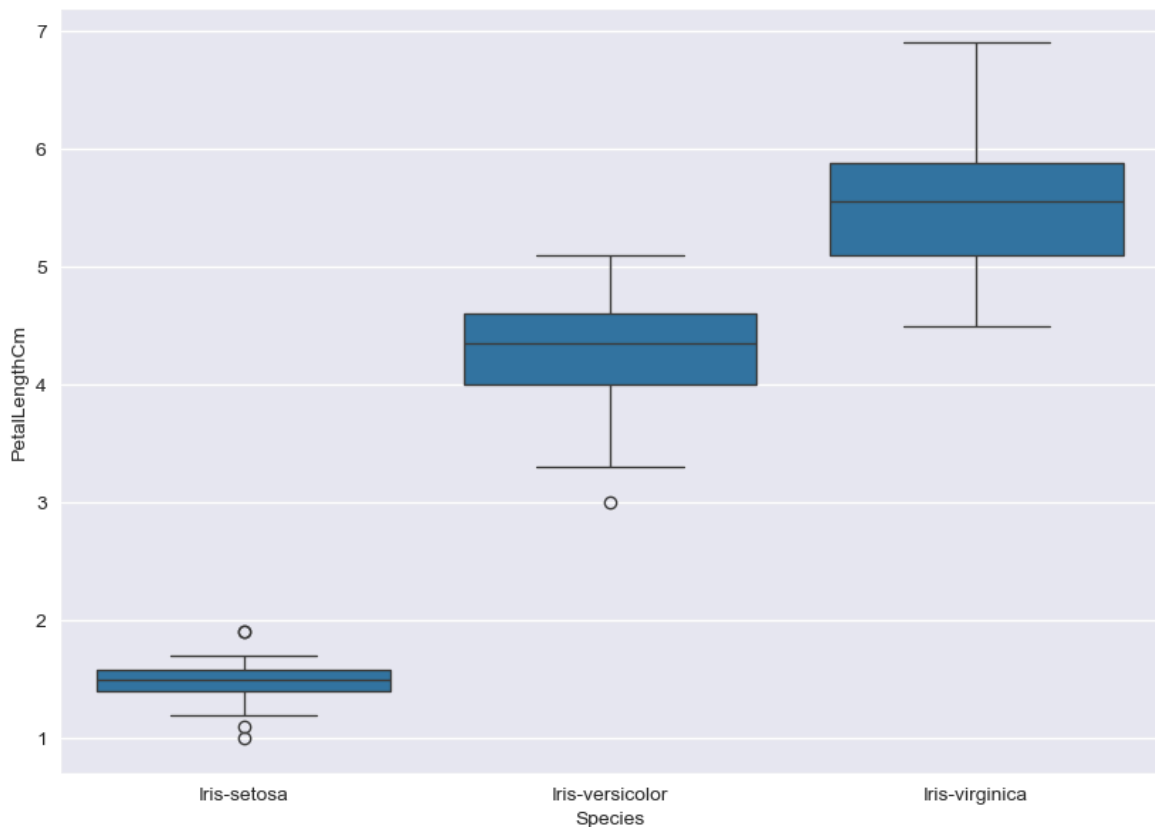
13.Swarm plot it looks a bit like a friendly swarm of bees buzzing about their hive . more importantly,each data point is clearly visible and no data are obscured by overplotting a beeswarm plot improves upon the random jittering approach to move data points the minimum distance away from one another to avoid overlays. the result is a plot where you can see each distinct data point like shown in below plot

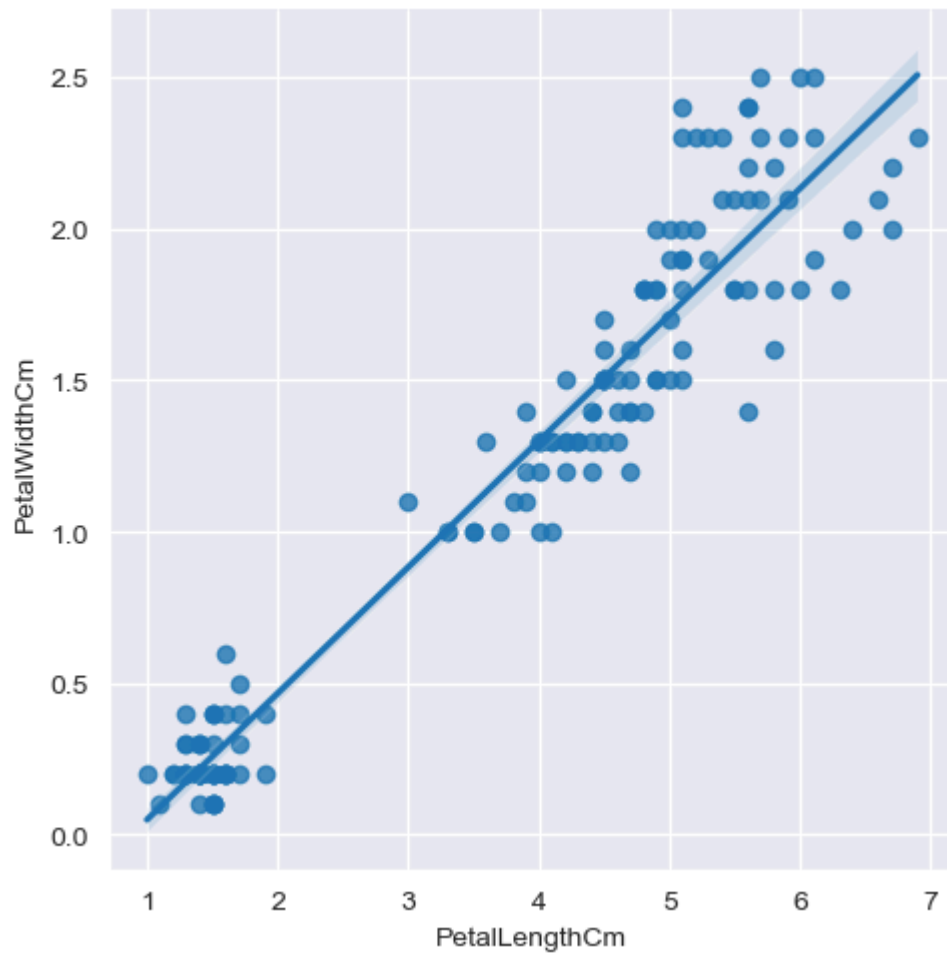
```
In [ ]: sns.set(style="darkgrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x="Species",y="PetalLengthCm", data=iris)
plt.show()
```

```
In [ ]: sns.set(style="whitegrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
ax = sns.violinplot(x="Species",y="PetalLengthCm", data= iris, inner = None)
ax= sns.swarmplot(x="Species",y="PetalLengthCm", data = iris, color= "white",edge
plt.show())
```

14. LM plot

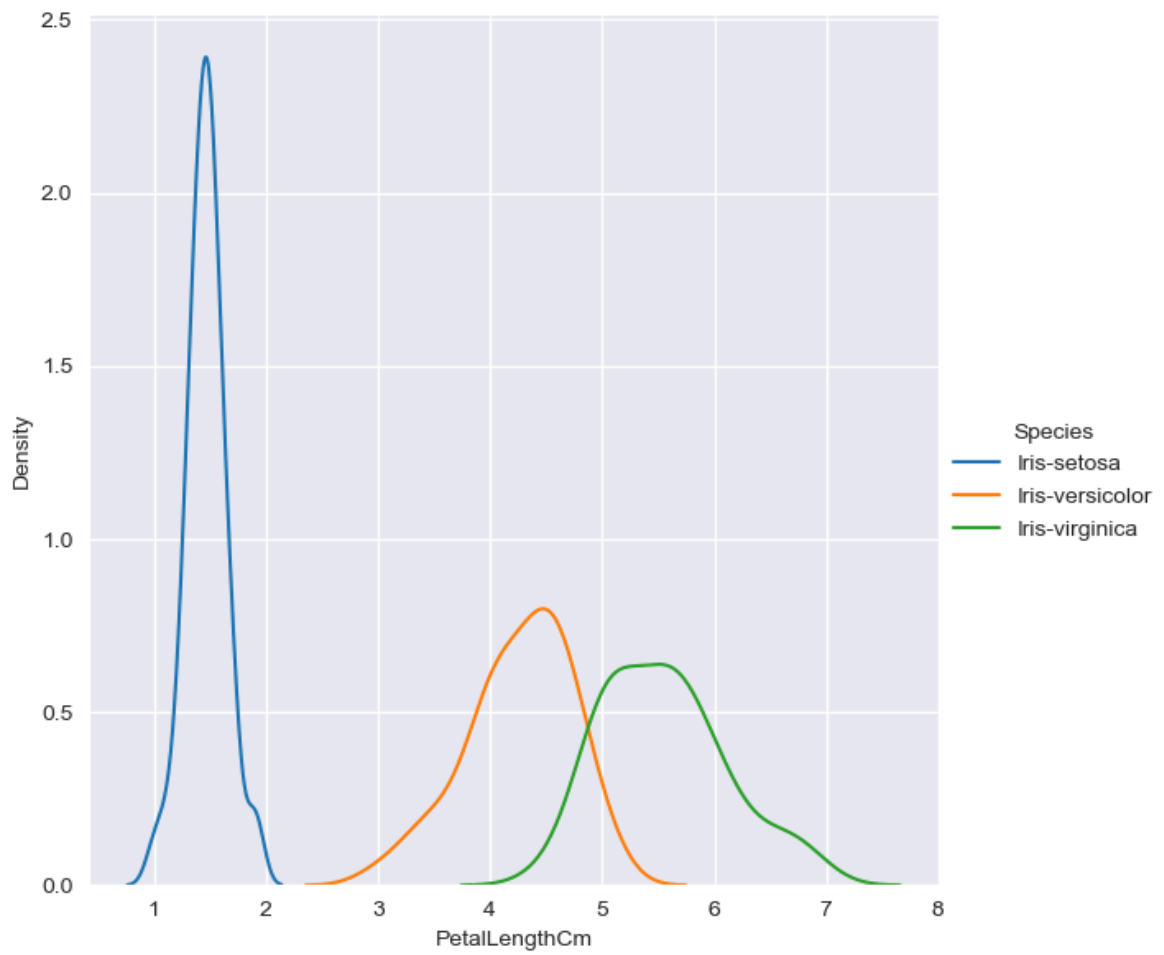
```
In [147... fig=sns.lmplot(x="PetalLengthCm",y="PetalWidthCm",data=iris)
plt.show()
```





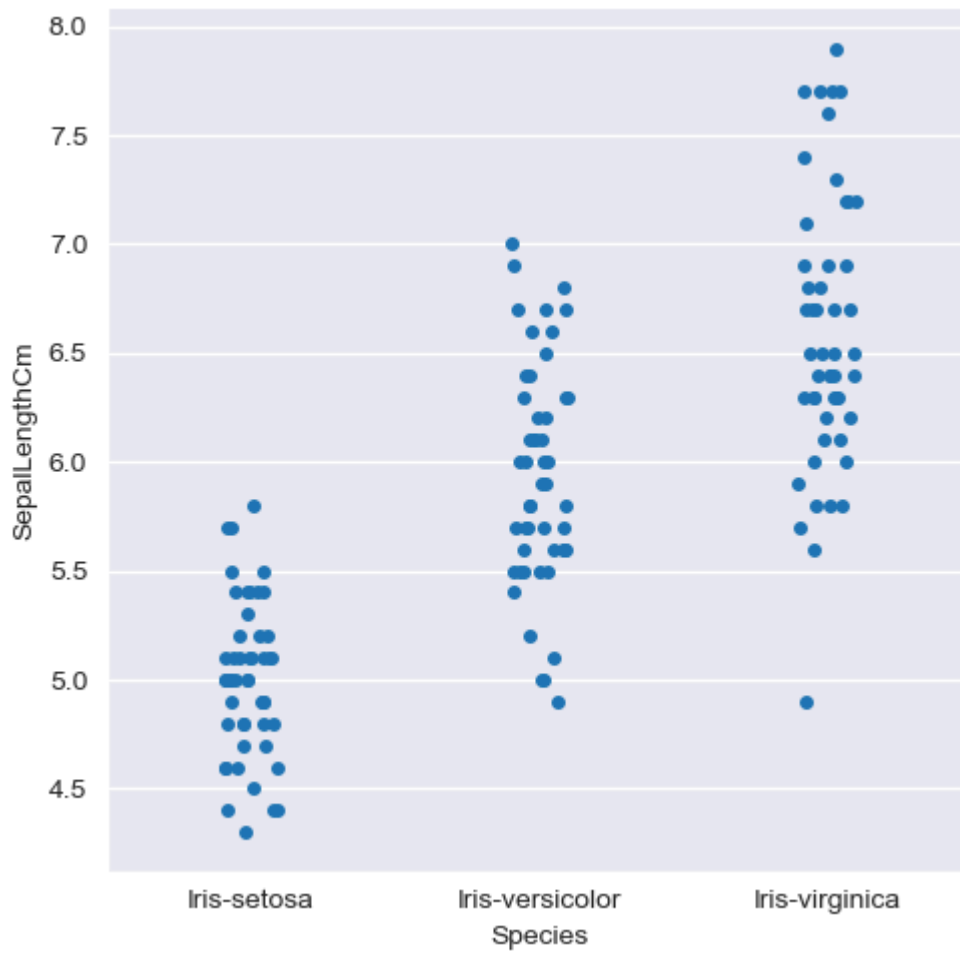
15.FacetGrid

```
In [149... sns.FacetGrid(iris, hue="Species", height=6) \
    .map(sns.kdeplot, "PetalLengthCm") \
    .add_legend()
plt.ioff()
plt.show()
```



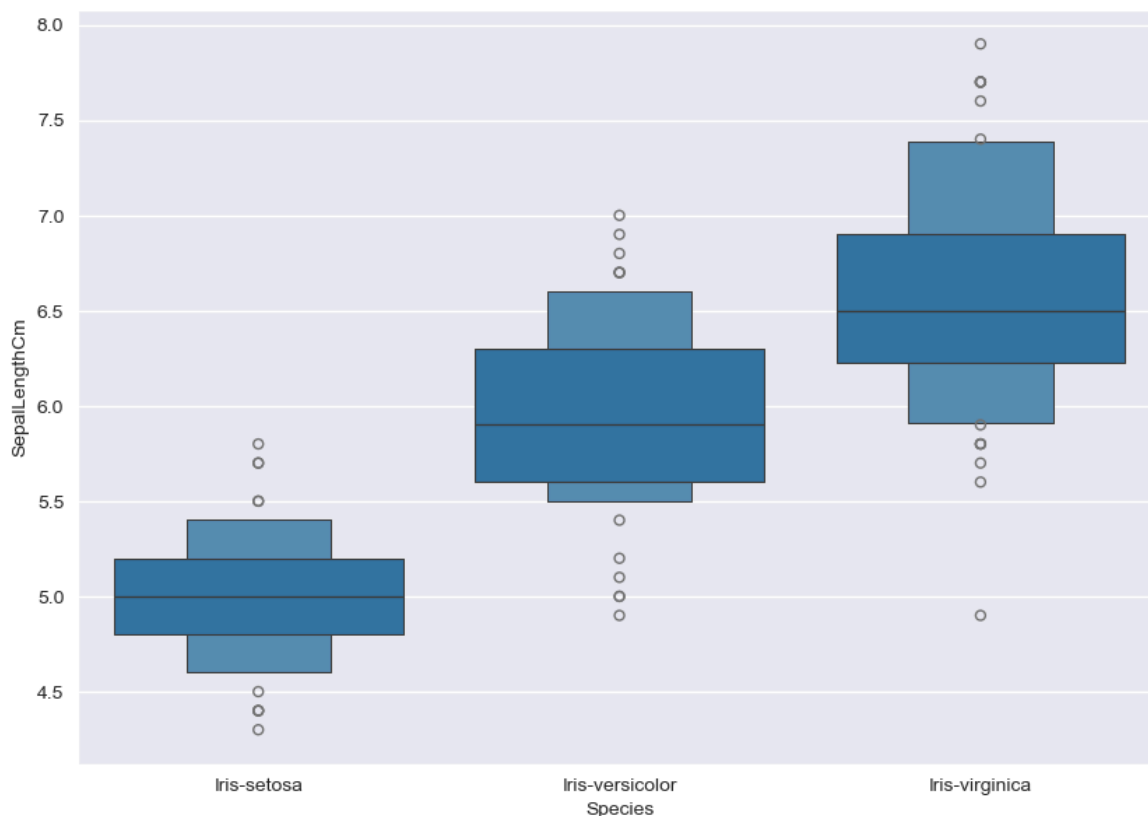
16. Factor plot

```
In [151... #f ax=plt.subplots(1,2,figsize=(18,8))
sns.catplot(x='Species',y='SepalLengthCm', data= iris)
plt.ioff()
plt.show()
```



17.Boxen plot

```
In [154... fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxenplot(x='Species',y='SepalLengthCm',data=iris)
plt.show()
```

In [155... `pip install --upgrade seaborn`

Requirement already satisfied: seaborn in c:\users\krishna\anaconda3\lib\site-packages (0.13.2)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\krishna\anaconda3\lib\site-packages (from seaborn) (1.26.4)

Requirement already satisfied: pandas>=1.2 in c:\users\krishna\anaconda3\lib\site-packages (from seaborn) (2.2.2)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\krishna\anaconda3\lib\site-packages (from seaborn) (3.9.2)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.0)

Requirement already satisfied: cycler>=0.10 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.51.0)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)

Requirement already satisfied: packaging>=20.0 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)

Requirement already satisfied: pillow>=8 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.2)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\krishna\anaconda3\lib\site-packages (from pandas>=1.2->seaborn) (2024.1)

Requirement already satisfied: tzdata>=2022.7 in c:\users\krishna\anaconda3\lib\site-packages (from pandas>=1.2->seaborn) (2023.3)

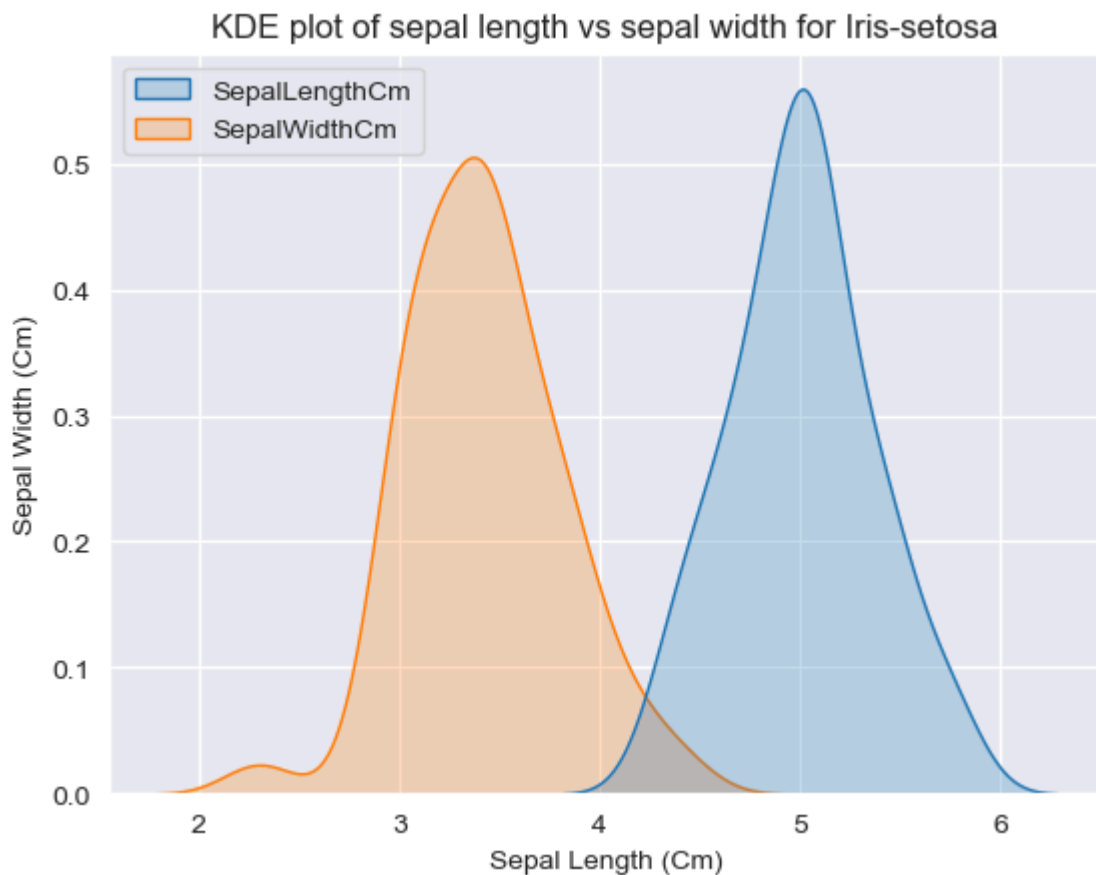
Requirement already satisfied: six>=1.5 in c:\users\krishna\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

Note: you may need to restart the kernel to use updated packages.

In []:

18. KDE plot

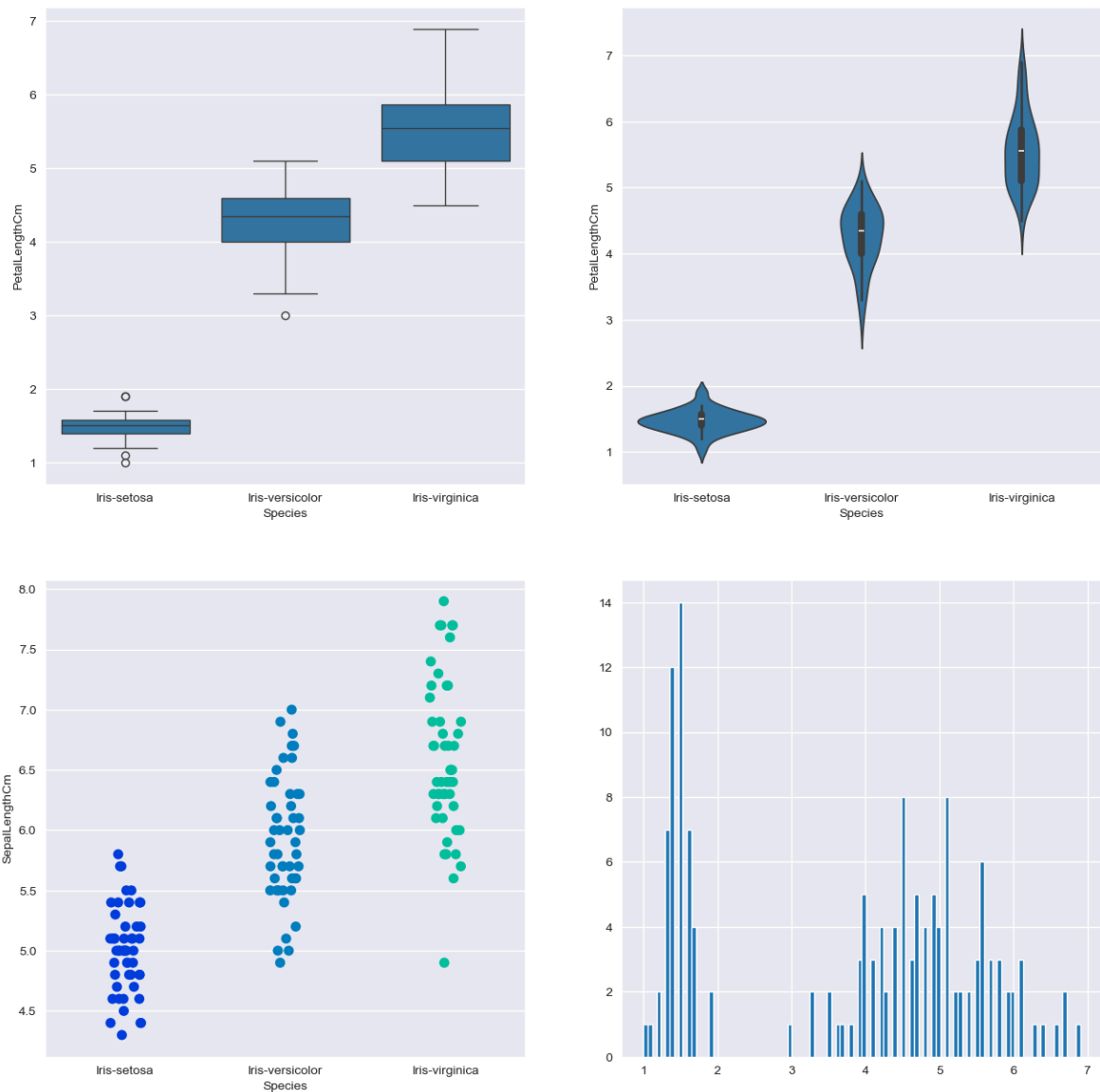
```
In [159... # Create a kde plot of sepal_length versus sepal width for setosa species of flo
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=sub[['SepalLengthCm','SepalWidthCm']], shade=True, shade_lowest
plt.title('KDE plot of sepal length vs sepal width for Iris-setosa')
plt.xlabel('Sepal Length (Cm)')
plt.ylabel('Sepal Width (Cm)')
plt.show()
```



19. Dashboard

```
In [161... sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.boxplot(x="Species", y="PetalLengthCm", data=iris,ax=axes[0,0])
k2=sns.violinplot(x='Species',y='PetalLengthCm',data=iris,ax=axes[0,1])
k3=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor='
#axes[1,1].hist(iris.hist,bin=10)
axes[1,1].hist(iris.PetalLengthCm,bins=100)
#k2.set(xlim=(-1,0.8))
plt.show()
```



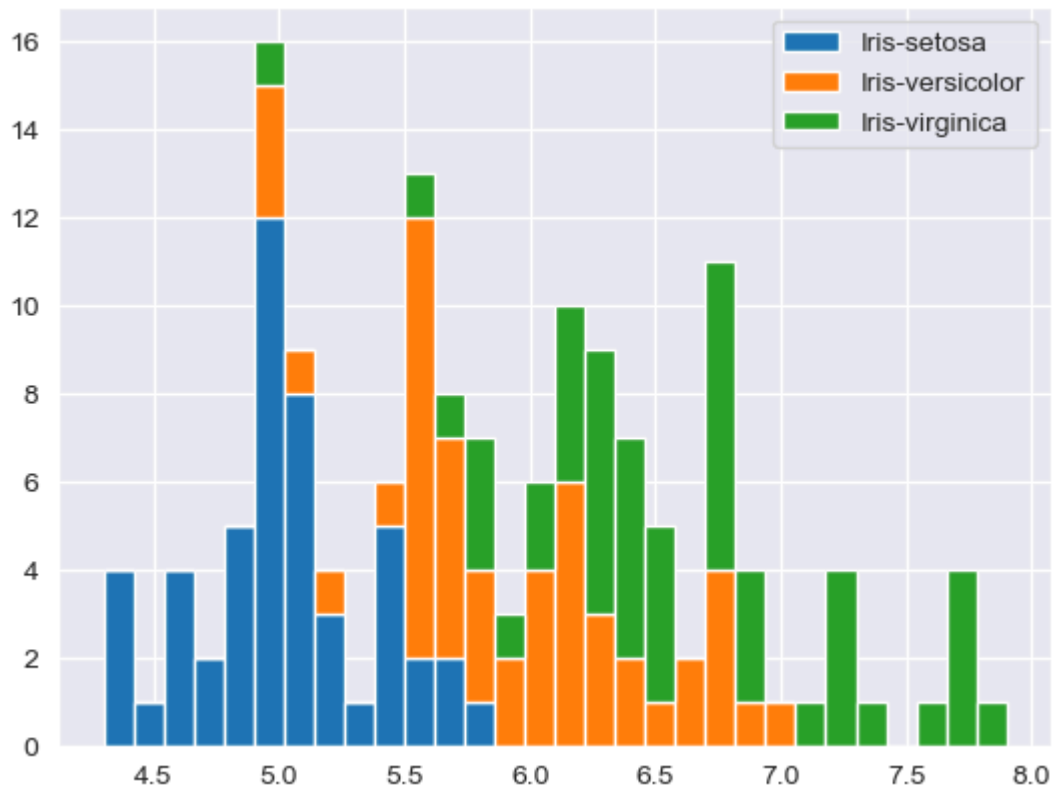
In this dashboard we have shown how to create multiple plots to form a dashboard using python in this plot we have demonstrated how to plot seaborn and matplotlib plots on the same dashboard.

20. Stacked Histogram

```
In [165...] iris['Species']=iris['Species'].astype('category')
```

```
In [167...] list1=list()
mylabels=list()
for gen in iris.Species.cat.categories:
    list1.append(iris[iris.Species==gen].SepalLengthCm)
    mylabels.append(gen)

h=plt.hist(list1,bins=30,stacked=True,rwidth=1,label=mylabels)
plt.legend()
plt.show()
```

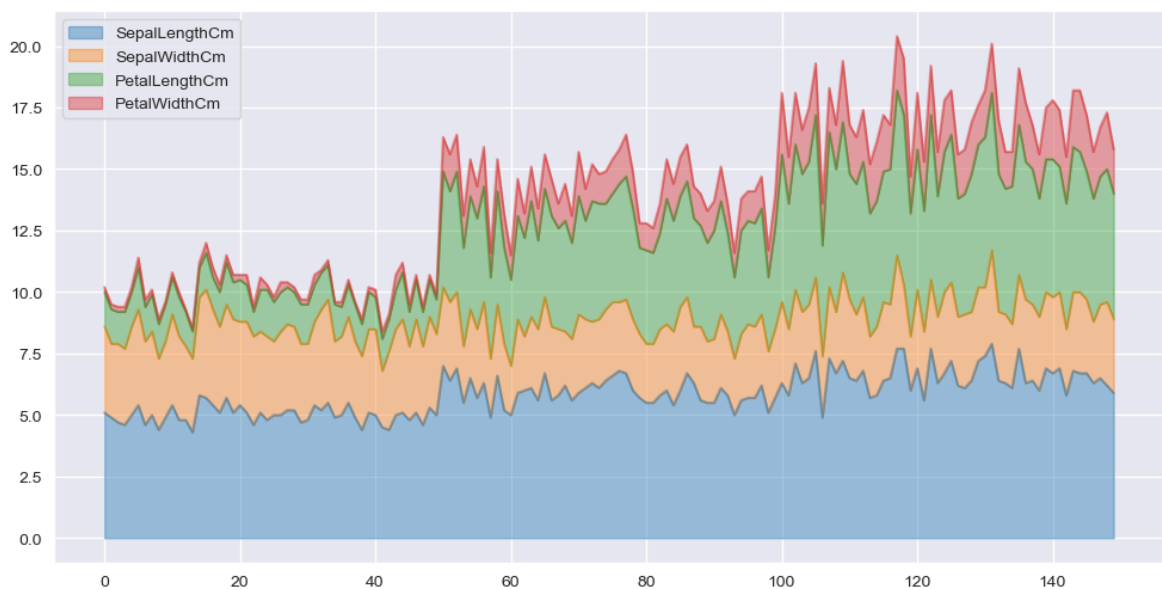


with stacked histogram we can see the distribution of sepal length of different species together. this shows us the range of sepal length for the three different species of iris flower.

21. Area Plot Area plot gives us a visual representation of various dimensions of iris flowers and their range in dataset,

In [170...

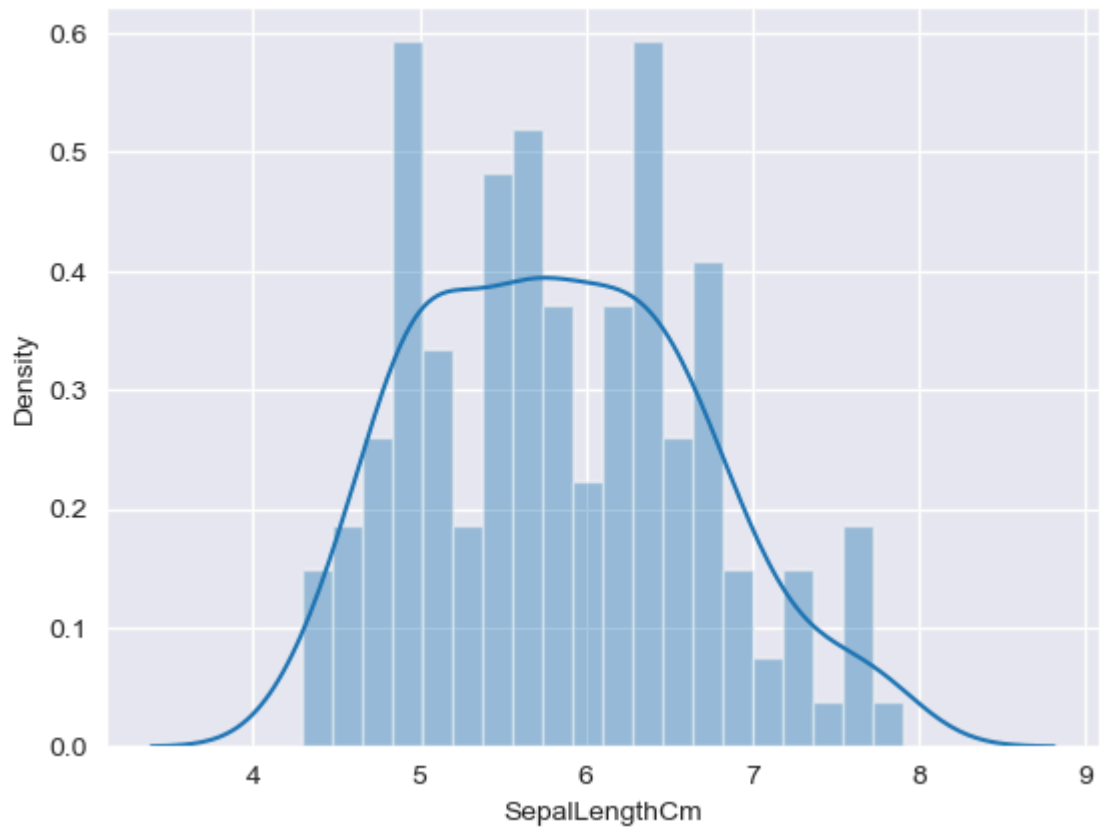
```
iris.plot.area(y=['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm'])
plt.show()
```



33. Distplot it helps us to look at the distribution of a single variable. KDE shows the density of the distribution

In [172...]

```
sns.distplot(iris['SepalLengthCm'],kde=True,bins=20);  
plt.show()
```



This is all about EDA ...>>>>

In []:

In []:

In []:

In []: