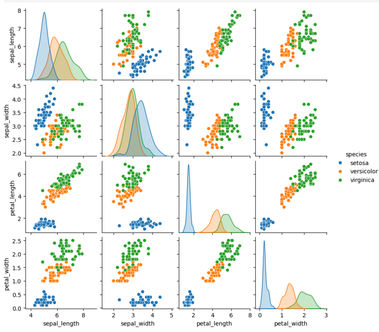
```
In [11]: import seaborn as sns
         import matplotlib.pyplot as plt
```

```
In [12]: #loading the iris dataset
         #to run this individual cell you need to be connected to internet
         iris = sns.load_dataset('iris')
```

# GENERAL STATISTICS PLOT

## PAIRPLOT

```
In [155]: # the below line of code creates a pairplot
          #hue is like legend in a graph or chart here it is (species)
          #height is to adjust the size of plot
          sns.pairplot(iris , hue ='species' , height=2.0)
          plt.show()
```




A pairplot is a set of scatterplots and distribution plots , distribution plots are along the diagonal
it helps in analysing any observable correlation or detect outliers or understand the distribution

## INTERPRETATION

-> from the scatter plots consider the 3,4 th columns we can see that irrespective of sepal length or width
setosa species petal length is mostly 1cm and petal width is mostly 0.5cm
-> in support for the upper statement we can consider the density plots in 3,4 th rows the distribution is spread
across (0-2) and (0-1) having more population at 1cm and 0.5cm respectively
-> we can find observable positive correlation between petal length and petal width irrespctive of species of the flower

## DESCRIBE

```
In [4]: iris.describe()
```

Out[4]:

| | sepal_length | sepal_width | petal_length | petal_width |
|---|---|---|---|---|
| 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 |

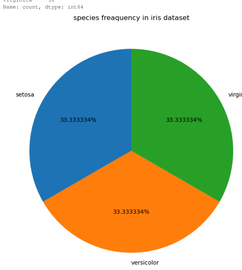
```
In [5]: iris.info()
```

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

## PIE CHART

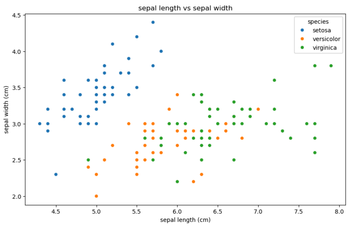
```
In [25]: species_counts = iris['species'].value_counts()
         print(species_counts)
         plt.figure(figsize=(8,8))
         #below statement is to create a pie chart labels is legend here
         #autopct parameter takes a formatted string as input this is to display the percentage of the element in the chart
         plt.pie(species_counts , labels=species_counts.index , autopct='%1.6f%%' , startangle=90)
         plt.title("species freaquency in iris dataset")
         plt.show()
```

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




## SCATTER PLOT

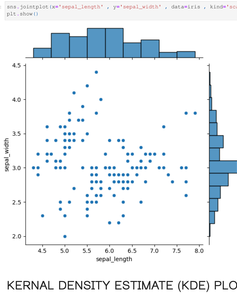
```
In [7]: plt.figure(figsize=(15,6))
        sns.scatterplot(x='sepal_length' , y='sepal_width' , hue='species' ,data=iris)
        plt.title('sepal length vs sepal width')
        plt.xlabel('sepal length (cm)')
        plt.ylabel('sepal width (cm)')
        plt.show()
```




## JOINT PLOT

a joint plot combines a scatterplot and a histogram
this plot can be used to find correlation as well as observable density of data points
-> for example in the below plot sepal length from 4.75 to 6.25 no of data points a nearer in this range most density is located
-> where as sepal width has more data points in the range 2.5 to 3.5 having high density

```
In [8]: sns.jointplot(x='sepal_length' , y='sepal_width' , data=iris , kind='scatter')
        plt.show()
```

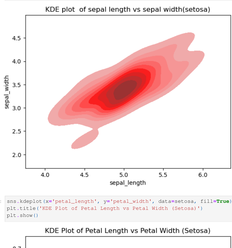



## KERNAL DENSITY ESTIMATE (KDE) PLOTS

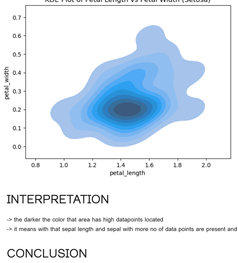
### BELOW KDE PLOTS ARE 2D KDE PLOTS

> BECAUSE WE USE TO DATA VARIABLES

```
In [30]: #here we are plotting on only setosa species so we are selecting only setosa from the iris data set
         setosa = iris[iris['species'] == 'setosa']
         #x,y  are two column inputs(sepal_length,petal_length) from the data(setosa)
         # fill is a special parameter with boolean value.In previous version it is named as shade and still can be used but not in upcoming versions
         # the results of the parameter are different range of the same color
         sns.kdeplot(x='sepal_length', y='sepal_width' , data=setosa , fill=True , color='r')
         plt.title('KDE plot  of sepal length vs sepal width(setosa)')
         plt.show()
```




```
In [10]: sns.kdeplot(x='petal_length', y='petal_width', data=setosa, fill=True)
         plt.title('KDE Plot of Petal Length vs Petal Width (Setosa)')
         plt.show()
```




## INTERPRETATION

-> the darker the color that area has high datapoints located
-> it means with that sepal length and sepal with more no of data points are present and same for petal length and petal width

## CONCLUSION

seaborn and matplotlib povides even more plots and features other than mentioned above
these packages hepls in both EDA using plots and graphs as well as to create some basic static dashboards