

```
In [28]: #flowers data set
#https://en.wikipedia.org/wiki/Iris_flower_data_set

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
In [19]: flowers_data1 = pd.read_csv(r"A:\LLM\flowers_data.CSV", encoding="ISO-8859-1")
flowers_data1.head()
```

```
Out[19]:
```

	Dataset order	Sepal length	Sepal width	Petal length	Petal width	Species
0	1	5.1	3.5	1.4	0.2	I. setosa
1	2	4.9	3.0	1.4	0.2	I. setosa
2	3	4.7	3.2	1.3	0.2	I. setosa
3	4	4.6	3.1	1.5	0.2	I. setosa
4	5	5.0	3.6	1.4	0.3	I. setosa

```
In [20]: flowers_data1.shape
```

```
Out[20]: (150, 6)
```

```
In [21]: flowers_data1.columns
```

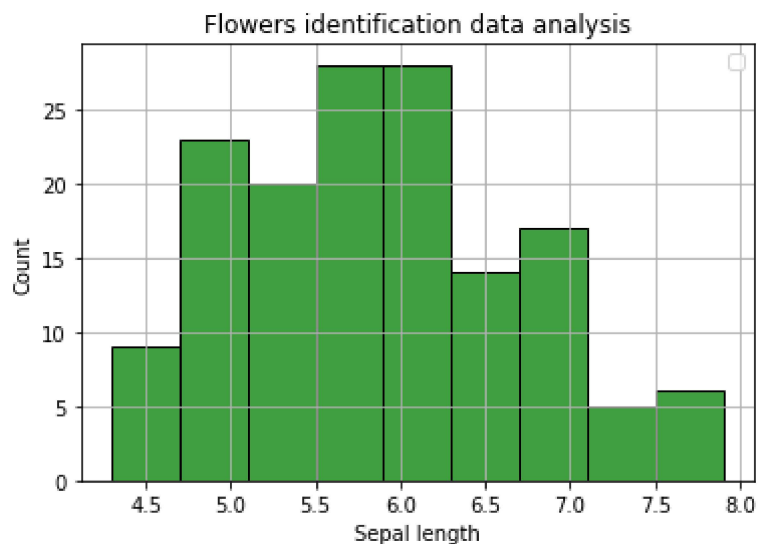
```
Out[21]: Index(['Dataset order', 'Sepal length', 'Sepal width', 'Petal length',
              'Petal width', 'Species'],
              dtype='object')
```

```
In [33]: #creating histplot with the help of seaborn library
#make sure here we are using both matplotlib lib and seaborn
#Like show(),grid() and title function only in matplotlib

sns.histplot(flowers_data1['Sepal length'], color='green')
plt.title("Flowers identification data analysis")
plt.grid()

#Yes! Seaborn can automatically group and aggregate data based on categorical column
#It does this using the hue, col, and row parameters in its plotting functions.
```

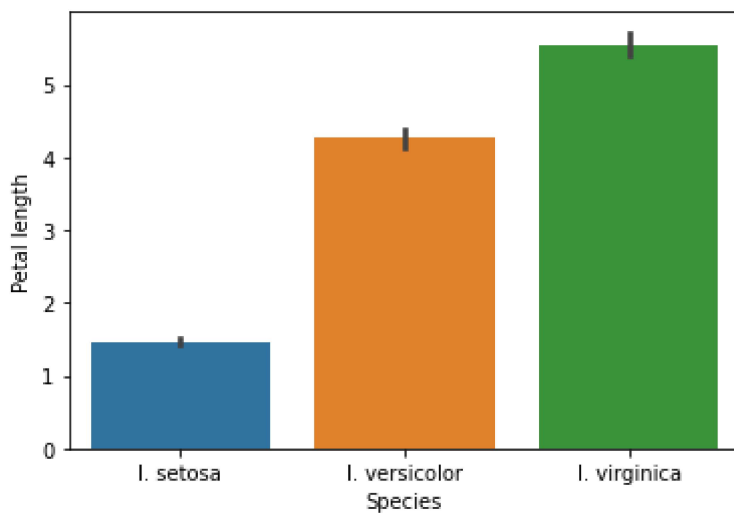
```
Out[33]: No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x25e593a7f70>
```



```
In [35]: #generatting barplot with help of seaborn

sns.barplot(x='Species', y='Petal length', data=flowers_data1)
```

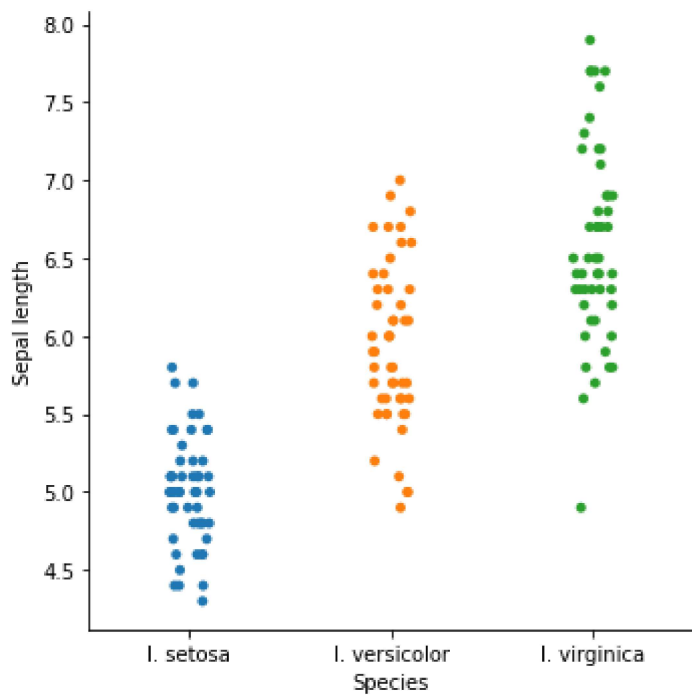
```
Out[35]: <AxesSubplot:xlabel='Species', ylabel='Petal length'>
```



```
In [36]: #generating catplot

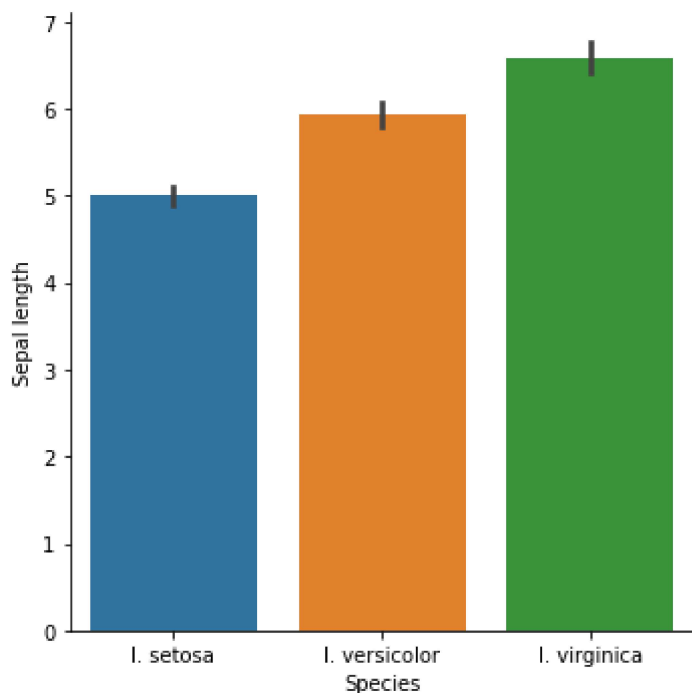
sns.catplot(x='Species', y='Sepal length', data=flowers_data1)
```

```
Out[36]: <seaborn.axisgrid.FacetGrid at 0x25e594e7580>
```



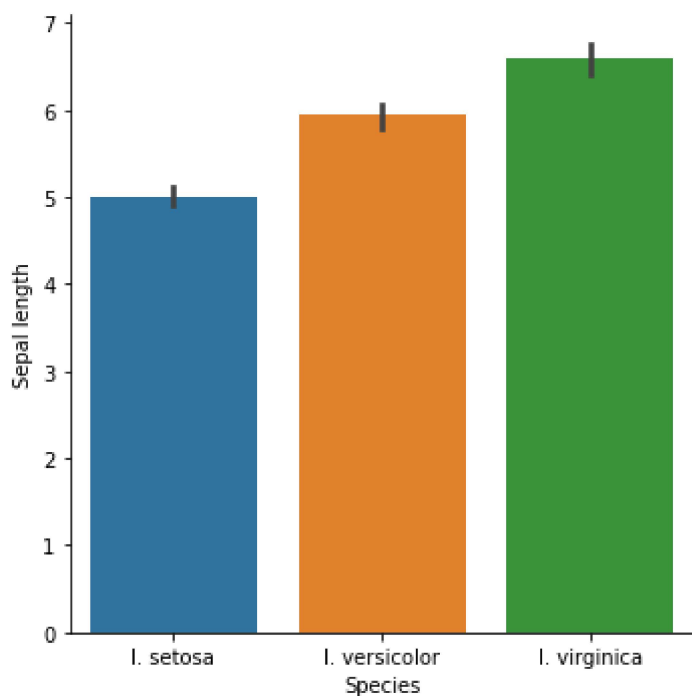
```
In [38]: #generating catplot , but here we are mentioning the kind ="bar"  
#so it is taking bar values  
  
sns.catplot(x='Species',y='Sepal length', data=flowers_data1, kind='bar')
```

```
Out[38]: <seaborn.axisgrid.FacetGrid at 0x25e5967da90>
```



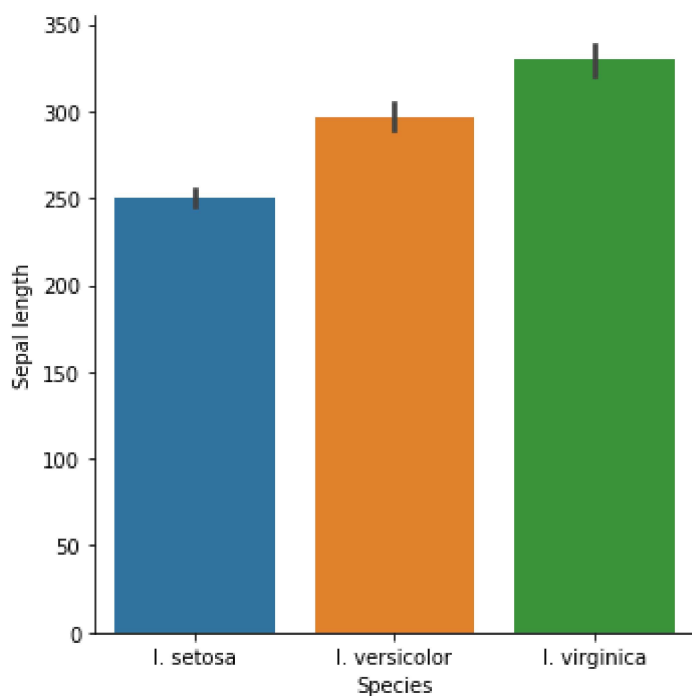
```
In [43]: sns.catplot(x='Species',y='Sepal length', data=flowers_data1, kind='bar', estimator=
```

```
Out[43]: <seaborn.axisgrid.FacetGrid at 0x25e5925aa00>
```



In [44]: `sns.catplot(x='Species',y='Sepal length', data=flowers_data1, kind='bar', estimator=`

Out[44]: `<seaborn.axisgrid.FacetGrid at 0x25e59518820>`



In [49]: `#value count giving data according to group by`

```
data2=v['Species'].value_counts()
data2
```

Out[49]:

I. setosa	50
I. versicolor	50
I. virginica	50

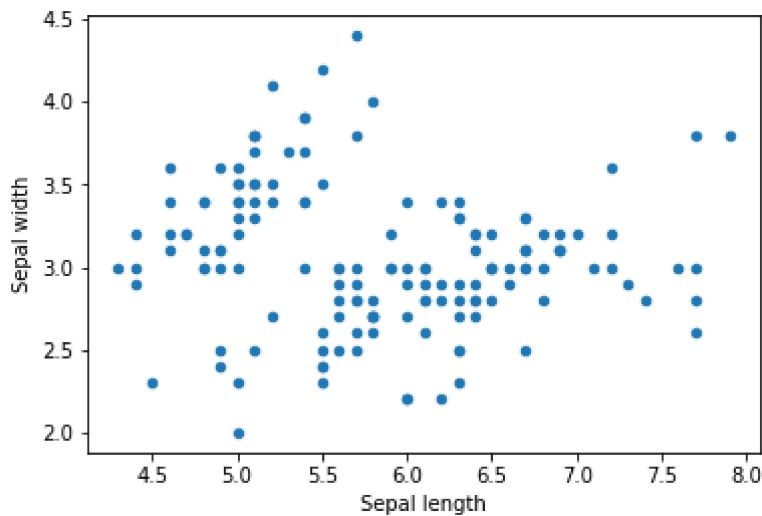
Name: Species, dtype: int64

In []: `#balanced data vs #unbalanced data`
`#here data is balanced because output of each category is 50`

```
#means if values are closed then balanced data
#if values are not closed then it is called unbalanced data
```

```
In [50]: flowers_data1.plot(kind='scatter',x='Sepal length' ,y='Sepal width')
```

```
Out[50]: <AxesSubplot:xlabel='Sepal length', ylabel='Sepal width'>
```



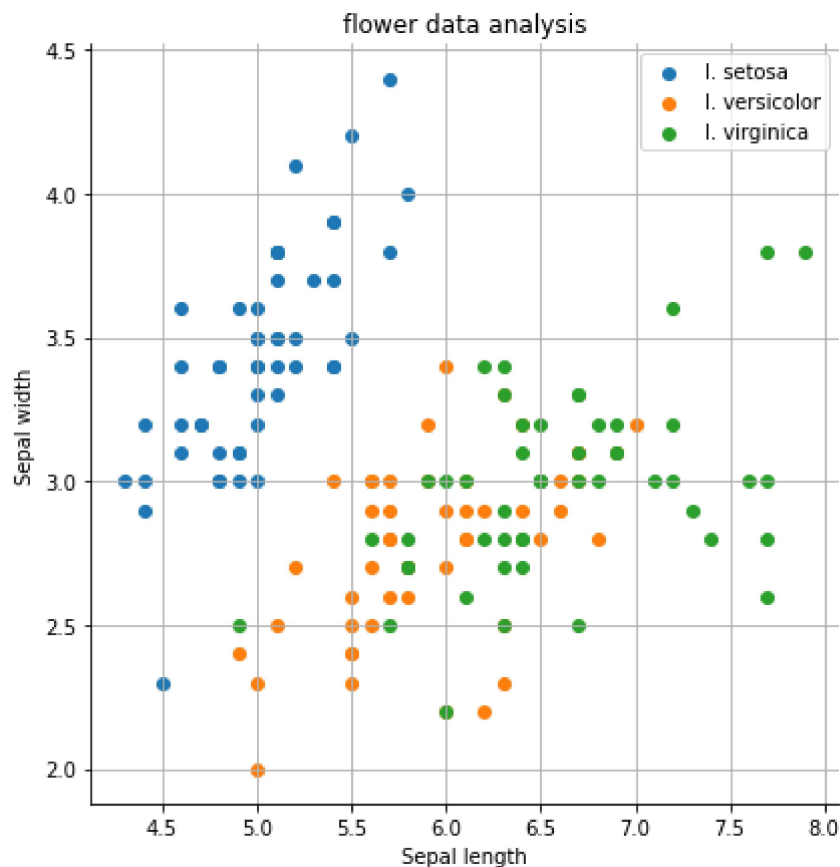
```
In [61]: #now we will design the same scatter graph with help of seaborn
#g = sns.FacetGrid(flowers_data1, hue="species", height=5)
sns.FacetGrid(flowers_data1,hue="Species",size=6) \
.map(plt.scatter,'Sepal length','Sepal width')
plt.grid()
plt.title("flower data analysis")
plt.legend()

#this is kind of two grpah
```

C:\Users\Gourav Sikka\anaconda3\lib\site-packages\seaborn\axisgrid.py:337: UserWarning: The `size` parameter has been renamed to `height`; please update your code.

warnings.warn(msg, UserWarning)

```
Out[61]: <matplotlib.legend.Legend at 0x25e5aca4160>
```



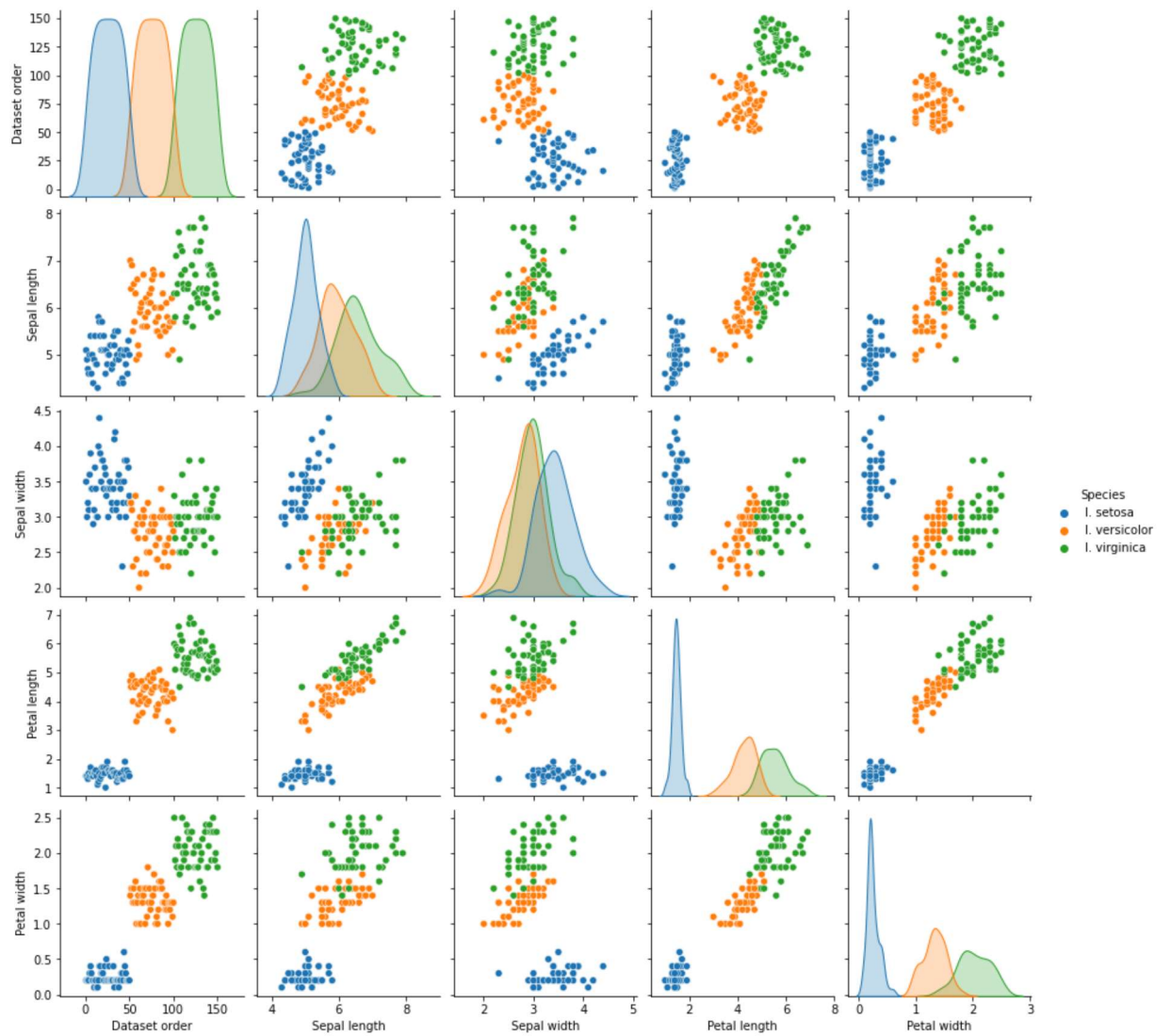
What is FacetGrid in Seaborn? FacetGrid is a powerful tool in Seaborn that allows you to create multiple subplots (facets) based on different categories in your dataset. It is useful when you want to visualize the distribution of data across multiple categories in a structured way.

In [63]:

```
#We can create a 1d , 2d and 3d graph only 4d and 5d or nd is not possible
#for creating a 3d graph we need to use seaborn but with

#via pairplot we can easily identification and see the pattren the distinguesh between

sns.pairplot(flowers_data1,hue="Species")
plt.grid()
```



In []:

In []:

In []: