Matplotlib

Data Visualization Technique

Markers

character	description
``'.'``	point marker
``','``	pixel marker
``'o'``	circle marker
``'v'``	triangle_down marker
``'^'``	triangle_up marker
``'<'``	triangle_left marker
``'>'``	triangle_right marker
``'1'``	tri_down marker
``'2'``	tri_up marker
``'3'``	tri_left marker
``'4'``	tri_right marker
``'8'``	octagon marker
``'s'``	square marker
``'p'``	pentagon marker
``'P'``	plus (filled) marker
``'*'``	star marker
``'h'``	hexagon1 marker
``'H'``	hexagon2 marker
``'+'``	plus marker
``'x'``	x marker
``'X'``	x (filled) marker
``'D'``	diamond marker
``'d'``	thin_diamond marker
``'|'``	vline marker
``'_'``	hline marker

Line Styles

character	description
``'-'``	solid line style
``'--'``	dashed line style
``'-.'``	dash-dot line style
``':'``	dotted line style

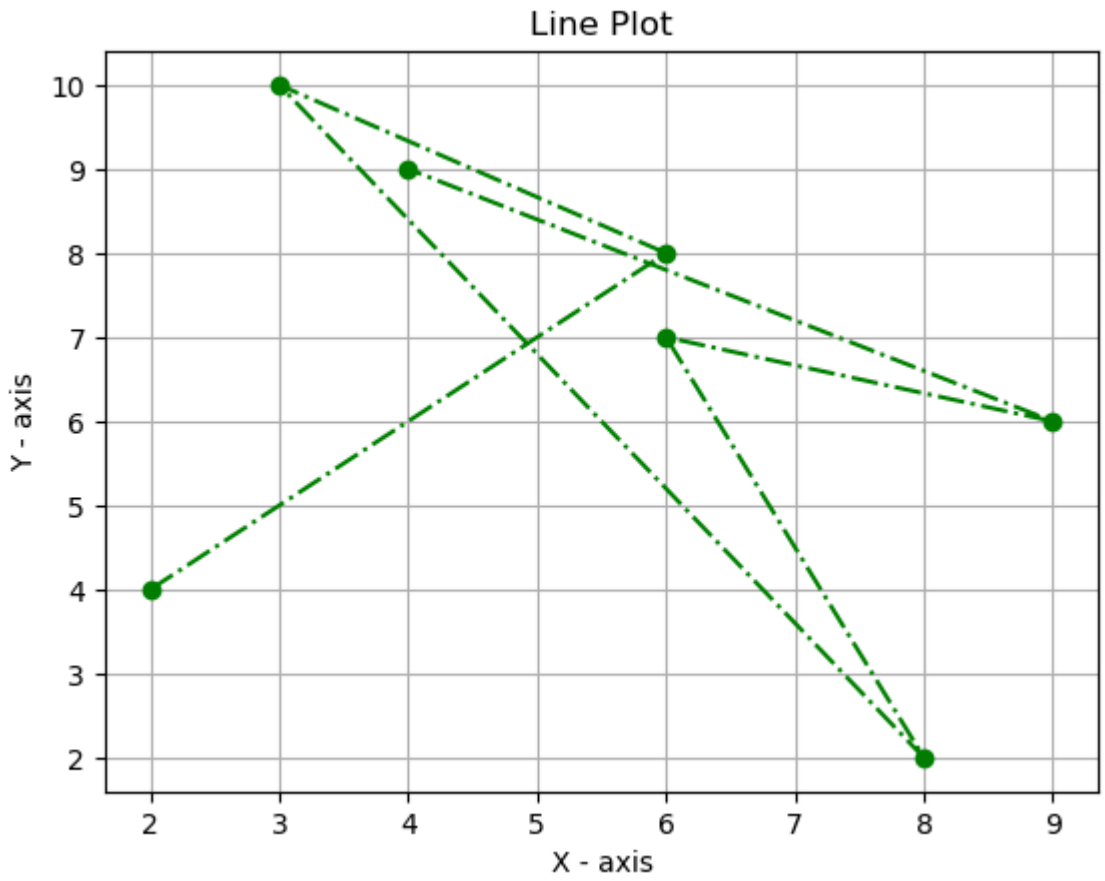
Example format strings::

```
    'b'    # blue markers with default shape
    'or'   # red circles
```

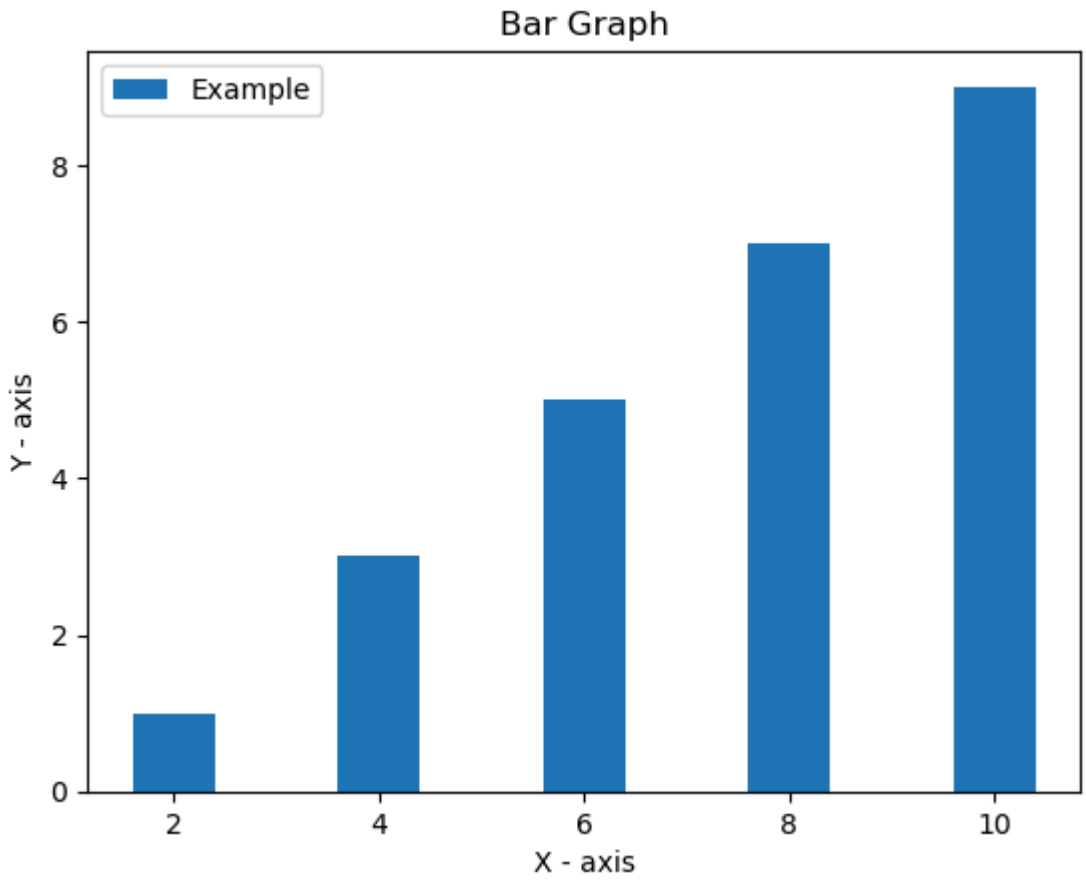
```
    '-g'   # green solid line
    '--'   # dashed line with default color
    '^k:'  # black triangle_up markers connected by a dotted
line

**Colors**

The supported color abbreviations are the single letter codes

character        color
=============    ================================
```'b'```        blue
```'g'```   green
```'r'```   red
```'c'```   cyan
```'m'```   magenta
```'y'```   yellow
```'k'```   black
```'w'```   white
```

In [52]:
```
import matplotlib.pyplot as plt
```

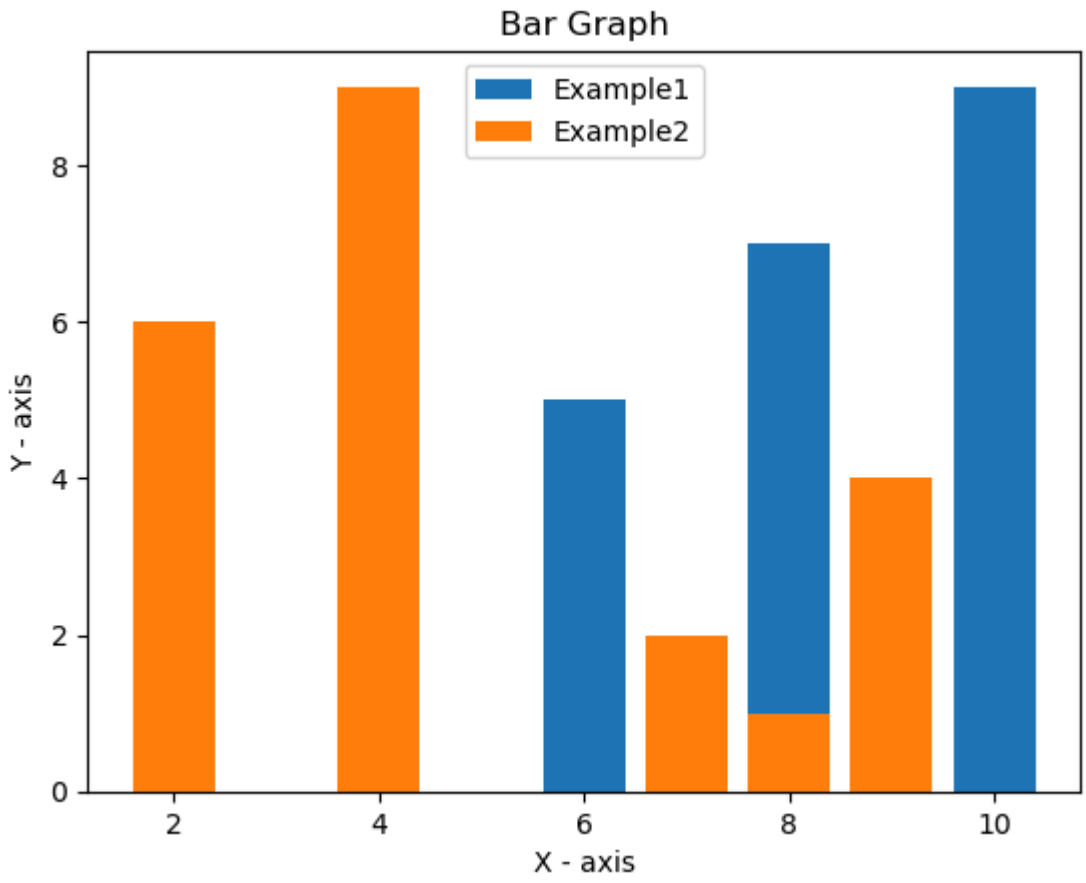
In [53]:
```
#Line Plot
x = [2,6,3,8,6,9,4]
y = [4,8,10,2,7,6,9]

plt.plot(x,y,'og-.')
plt.title('Line Plot')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.grid()
plt.show()
```



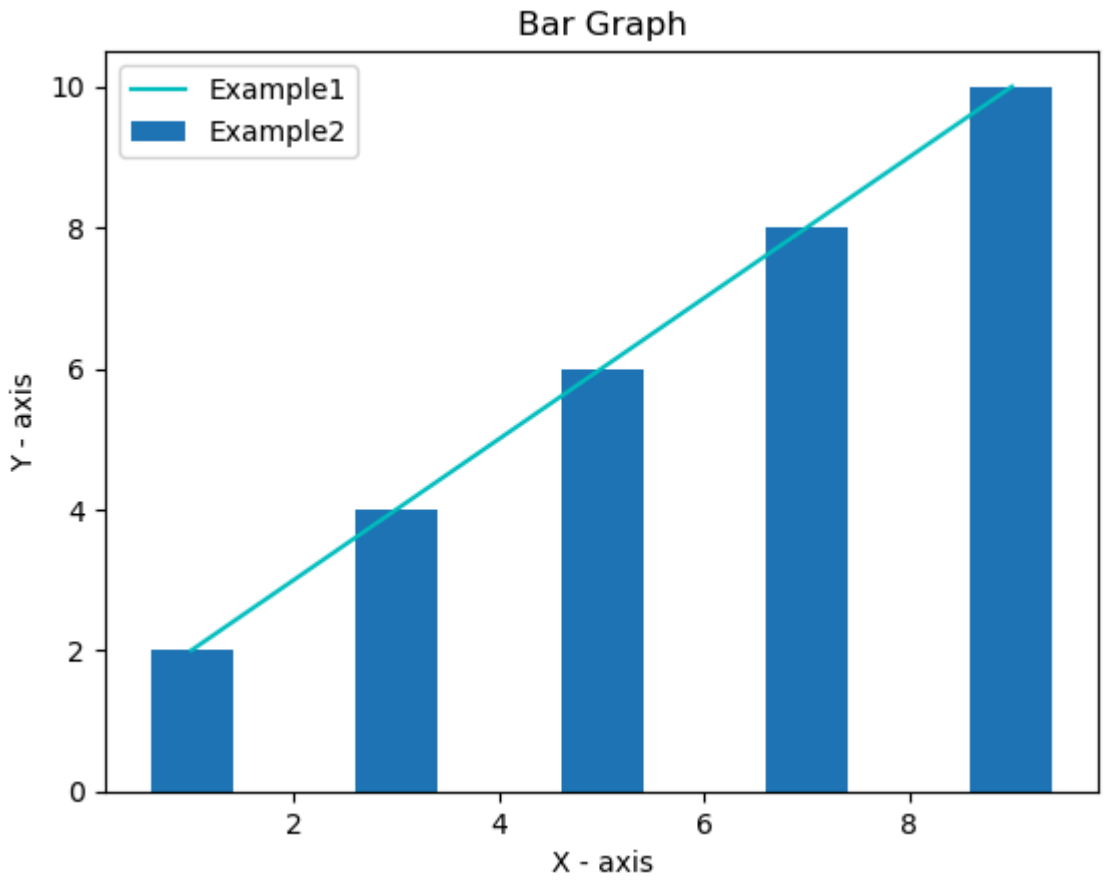
In [3]:
```
#Bar Plot
x = [2,4,6,8,10]
y = [1,3,5,7,9]

plt.bar(x,y,label='Example')
plt.title('Bar Graph')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.legend()
plt.show()
```



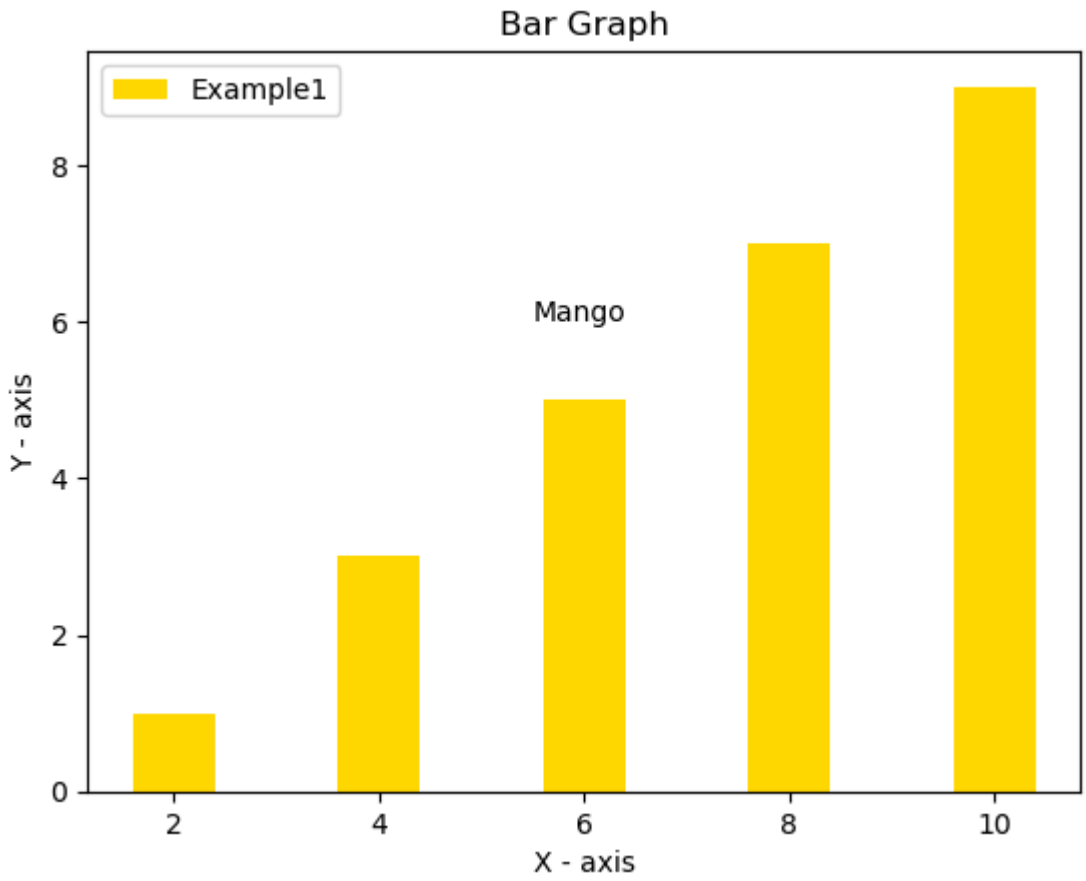
In [4]:
```
x = [2,4,6,8,10]
y = [1,3,5,7,9]
plt.bar(x,y,label='Example1')

x1 = [4,7,2,9,8]
y1 = [9,2,6,4,1]
plt.bar(x1,y1,label='Example2')

plt.title('Bar Graph')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.legend()
plt.show()
```



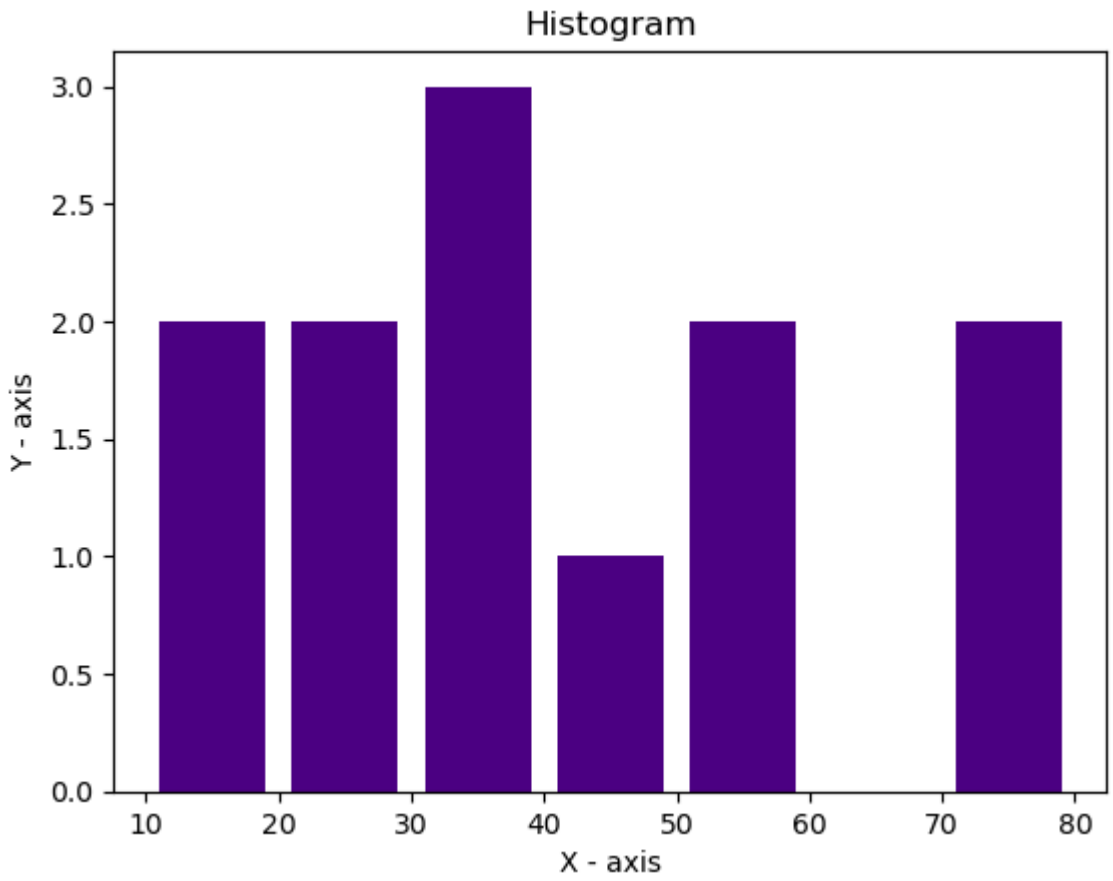
In [5]:
```
x = [1,3,5,7,9]
y = [2,4,6,8,10]
plt.plot(x,y,label='Example1',color='c')

x1 = [1,3,5,7,9]
y1 = [2,4,6,8,10]
plt.bar(x1,y1,label='Example2')

plt.title('Bar Graph')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.legend()
plt.show()
```



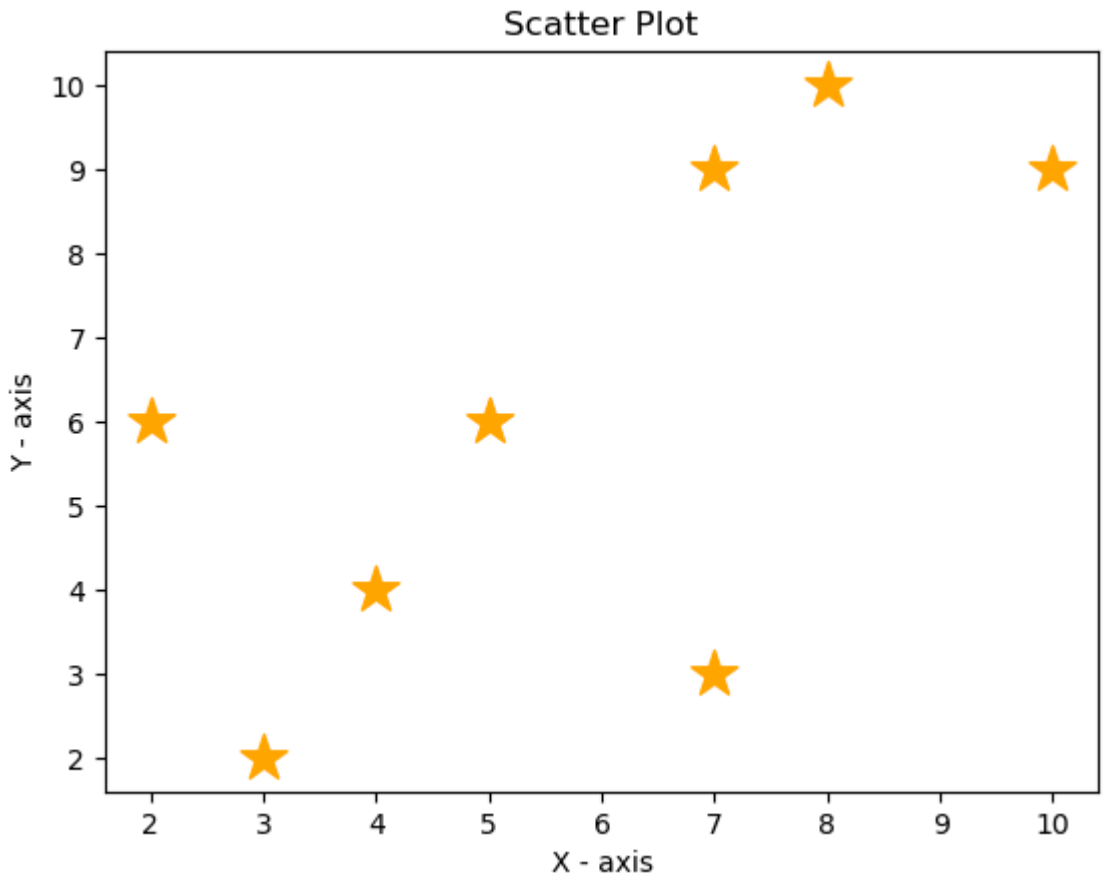
In [6]:
```
x = [2,4,6,8,10]
y = [1,3,5,7,9]
plt.bar(x,y,label='Example1',color='gold')

plt.title('Bar Graph')
plt.annotate('Mango',xy=(5.5,6))
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.legend()
plt.show()
```



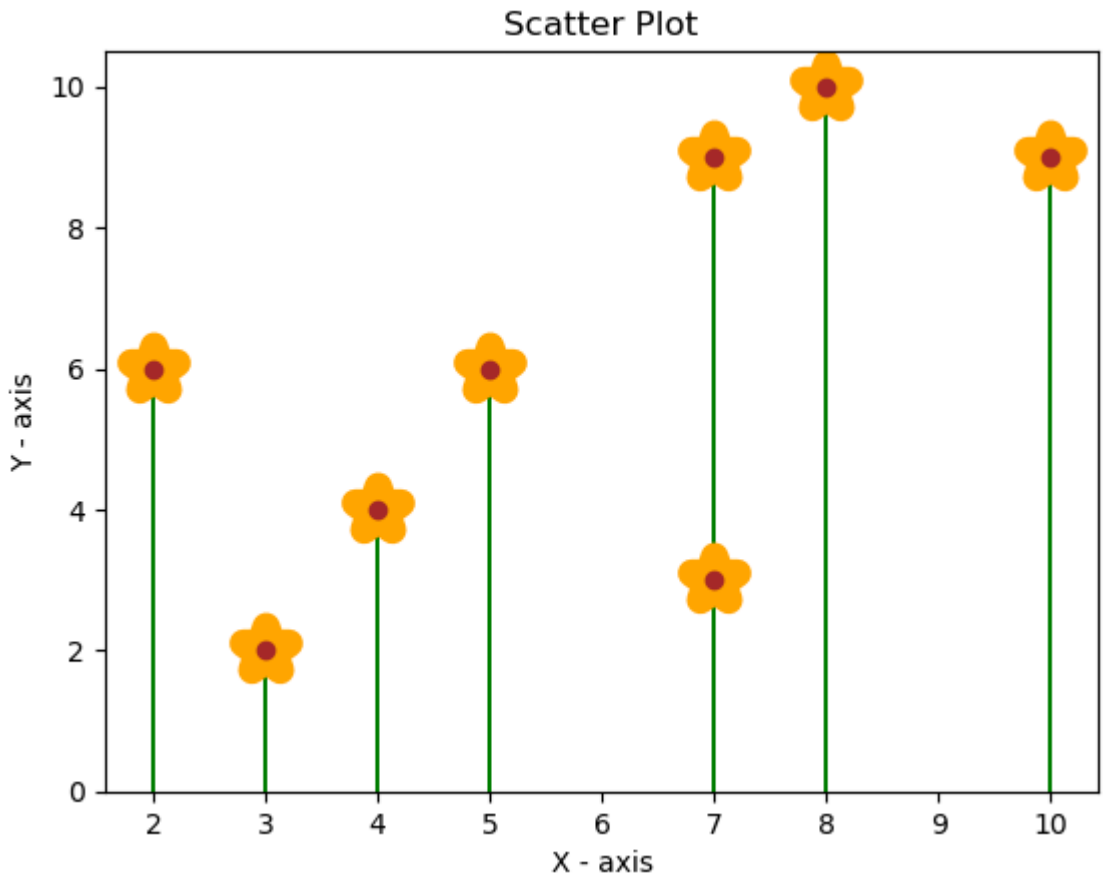
In [7]:
```
# Histogram
ages = [20,40,56,78,34,23,6,53,78,12,35,10,30]
bins = [10,20,30,40,50,60,70,80]

plt.hist(ages,bins,histtype='bar',rwidth=0.8, color='indigo')
plt.title('Histogram')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.show()
```



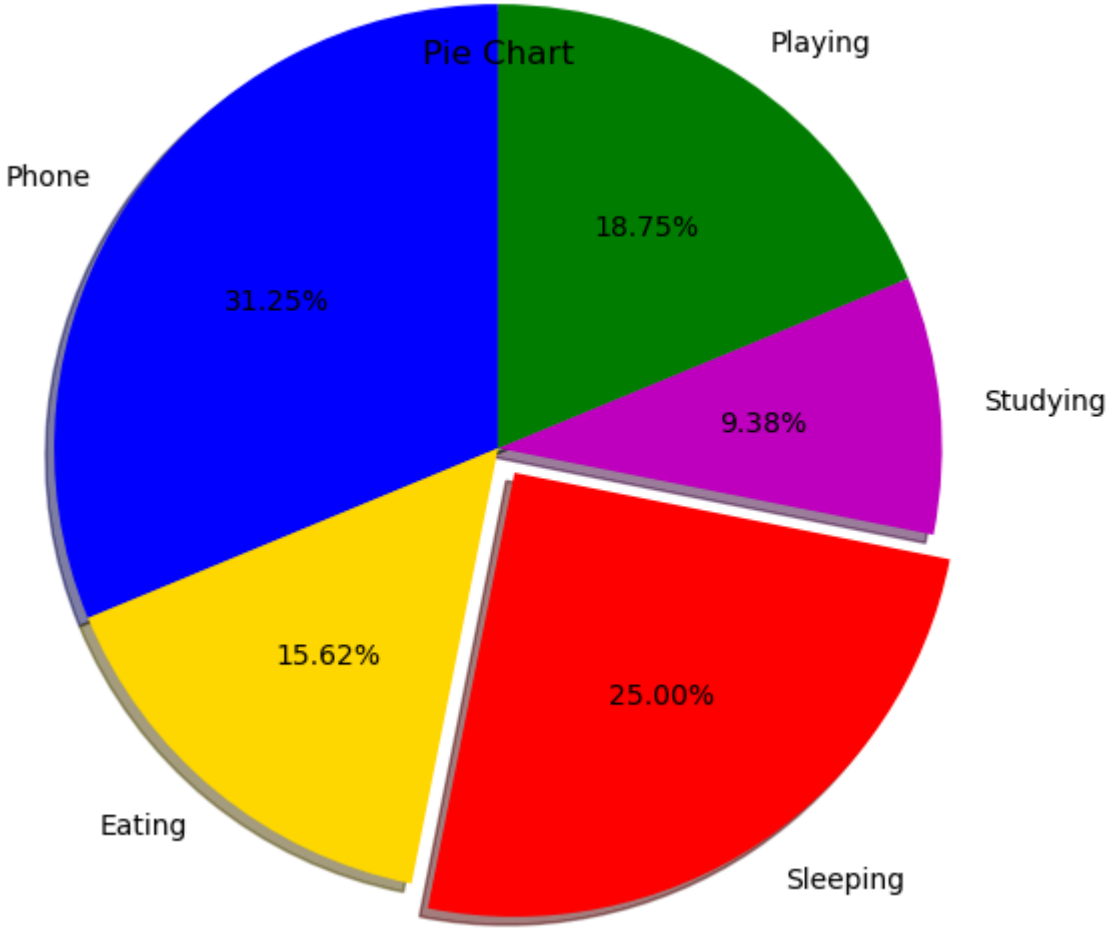
In [56]:
```
#Scatter Plot
x1 = [3,5,4,7,2,10,7,8]
y1 = [2,6,4,9,6,9,3,10]

plt.scatter(x1,y1,s=300,marker='*',color='orange')
plt.title('Scatter Plot')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.show()
```



In [57]:
```
#Scatter Plot & Bar Plot
x2 = [3,5,4,7,2,10,7,8]
y2 = [2,6,4,9,6,9,3,10]
plt.bar(x2,y2,width=0.05,color='green')

x = [3,5,4,7,2,10,7,8]
y = [2,6,4,9,6,9,3,10]
plt.scatter(x,y,s=300,color='gold',marker='*',edgecolors='orange',linewidths=10)

x1 = [3,5,4,7,2,10,7,8]
y1 = [2,6,4,9,6,9,3,10]
plt.scatter(x1,y1,color='brown')

plt.title('Scatter Plot')
plt.xlabel('X - axis')
plt.ylabel('Y - axis')
plt.show()
```



In [9]:
```
# Pie Chart
slices = [10,5,8,3,6]
activities = ['Phone','Eating','Sleeping','Studying','Playing']
cols = ['b','gold','r','m','g']

plt.pie(slices,labels=activities,colors=cols,startangle=90,shadow=True,
       explode=(0,0,0.1,0,0),autopct='%1.2f%%',radius=1.5)
plt.title('Pie Chart')
plt.show()
```



Seaborn

Statistical Data Visualization Technique

In [12]:
```
#Import required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:
```
#to ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

In [14]:
```
iris = pd.read_csv(r"C:\Users\Lab-26\Downloads\iris (3)\iris.data",header=None)
iris
```

Out[14]:

	0	1	2	3	4
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [15]:
```
iris.columns = ['SL','SW','PL','PW','Flower']
iris.head()
```

Out[15]:

	SL	SW	PL	PW	Flower
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

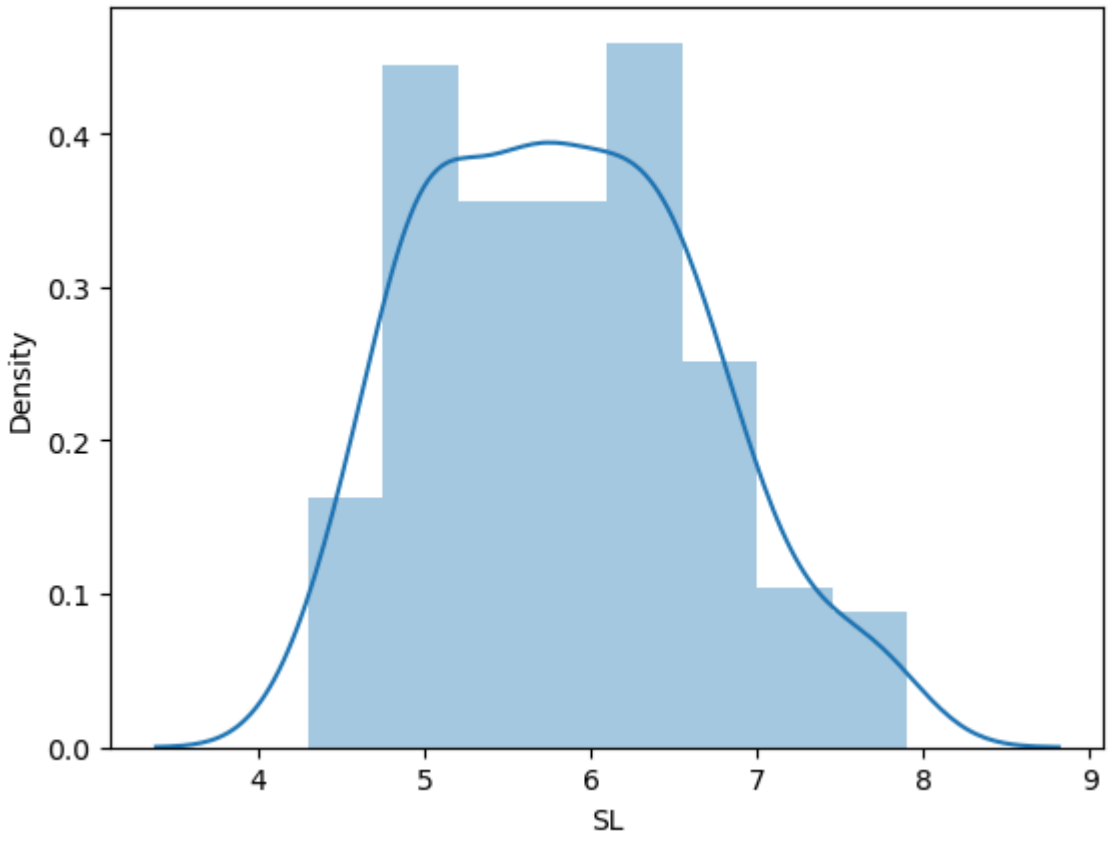
In [16]:
```
iris.isnull().sum()
```

Out[16]:
```
SL         0
SW         0
PL         0
PW         0
Flower     0
dtype: int64
```

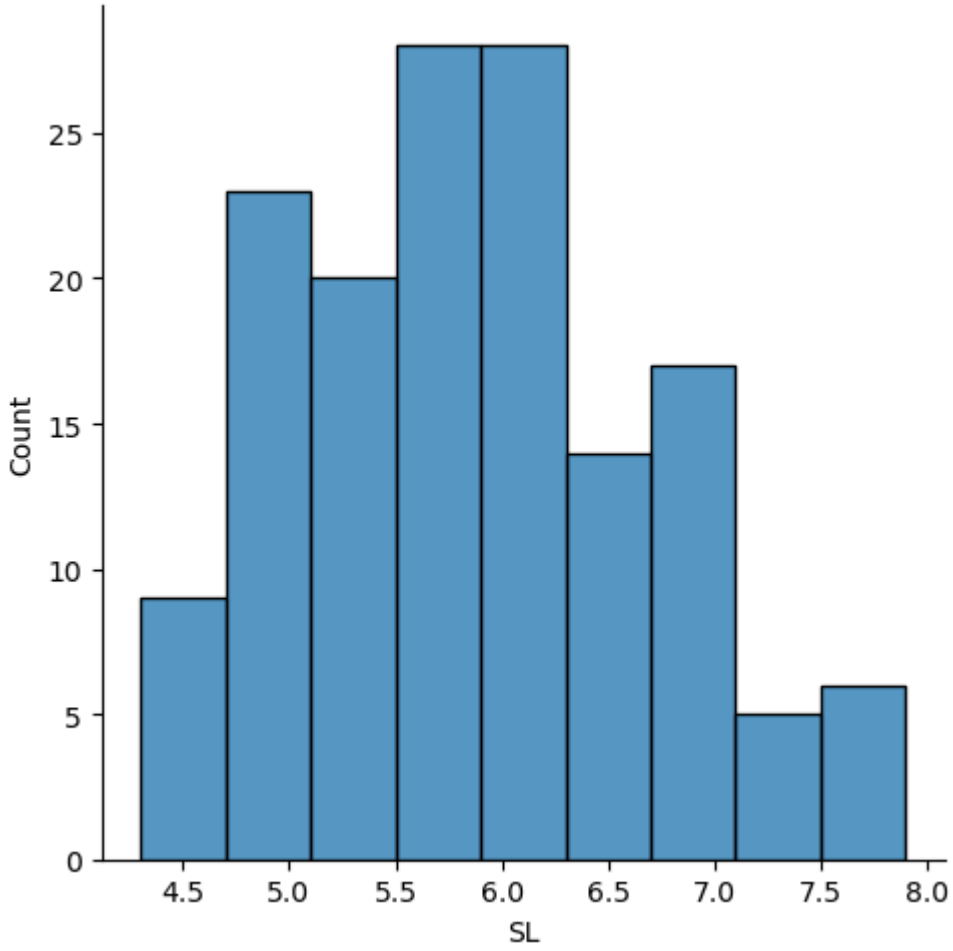
In [17]:
```
for i in iris.columns:
    print(i,':','\n',iris[i].unique(),'\n')
```

```
SL :
 [5.1 4.9 4.7 4.6 5.  5.4 4.4 4.8 4.3 5.8 5.7 5.2 5.5 4.5 5.3 7.  6.4 6.9
 6.5 6.3 6.6 5.9 6.  6.1 5.6 6.7 6.2 6.8 7.1 7.6 7.3 7.2 7.7 7.4 7.9]

SW :
 [3.5 3.  3.2 3.1 3.6 3.9 3.4 2.9 3.7 4.  4.4 3.8 3.3 4.1 4.2 2.3 2.8 2.4
 2.7 2.  2.2 2.5 2.6]

PL :
 [1.4 1.3 1.5 1.7 1.6 1.1 1.2 1.  1.9 4.7 4.5 4.9 4.  4.6 3.3 3.9 3.5 4.2
 3.6 4.4 4.1 4.8 4.3 5.  3.8 3.7 5.1 3.  6.  5.9 5.6 5.8 6.6 6.3 6.1 5.3
 5.5 6.7 6.9 5.7 6.4 5.4 5.2]

PW :
 [0.2 0.4 0.3 0.1 0.5 0.6 1.4 1.5 1.3 1.6 1.  1.1 1.8 1.2 1.7 2.5 1.9 2.1
 2.2 2.  2.4 2.3]

Flower :
 ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

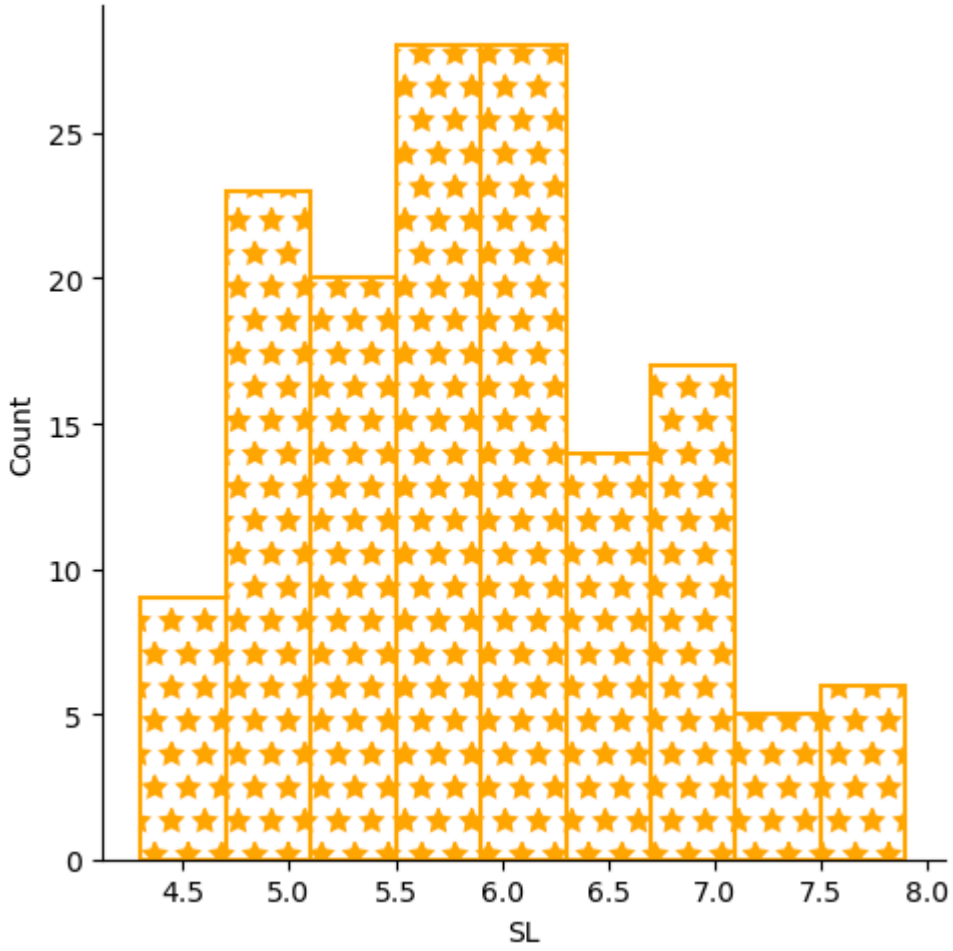
In [18]:
```
#Distribution Plot
sns.distplot(iris.SL)
plt.show()
```



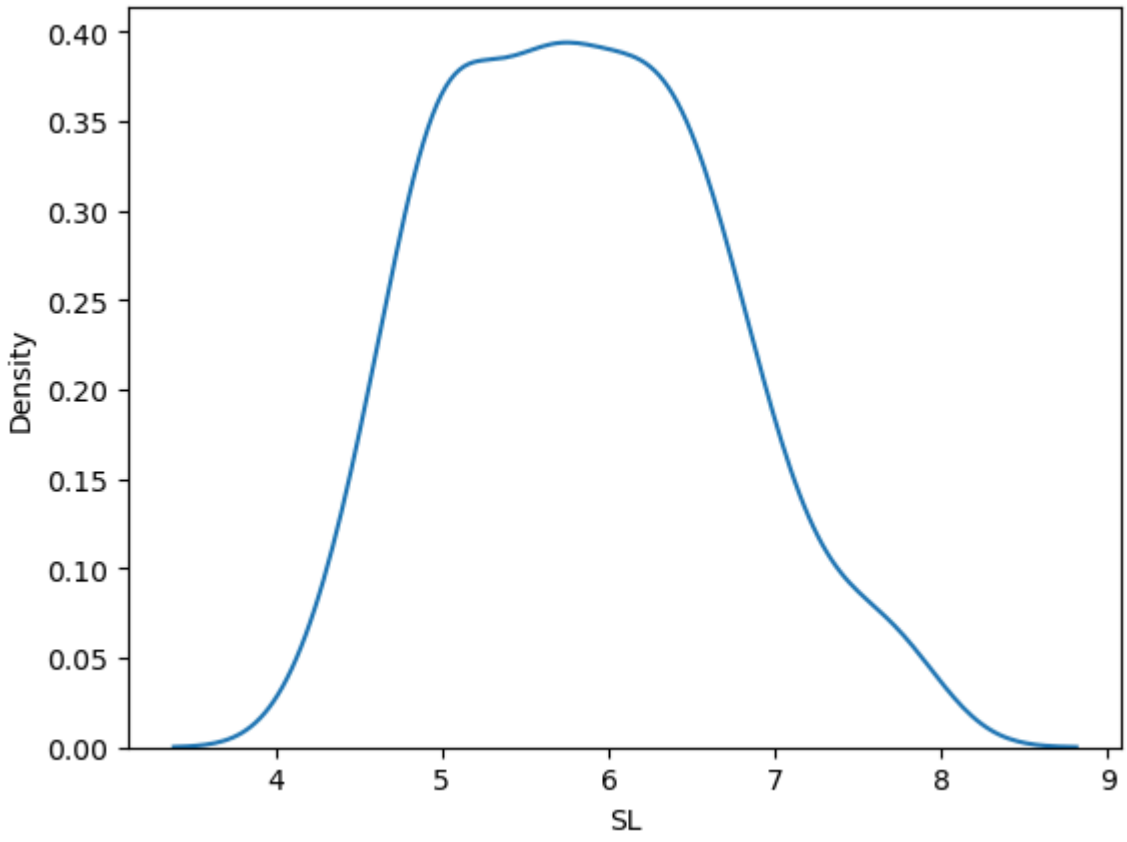
In [19]:
```
sns.displot(iris.SL)
plt.show()
```



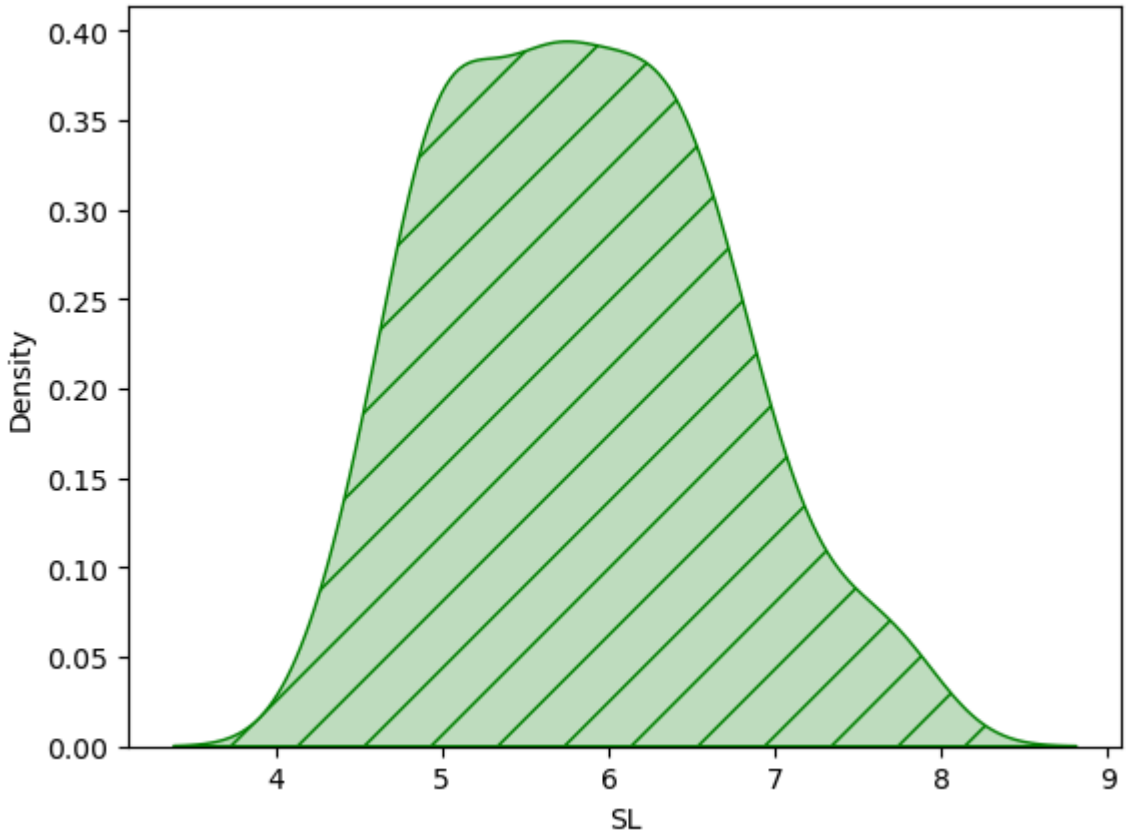
In [20]:
```
sns.displot(iris.SL,fill=False,hatch='*',color='orange')
plt.show()
```



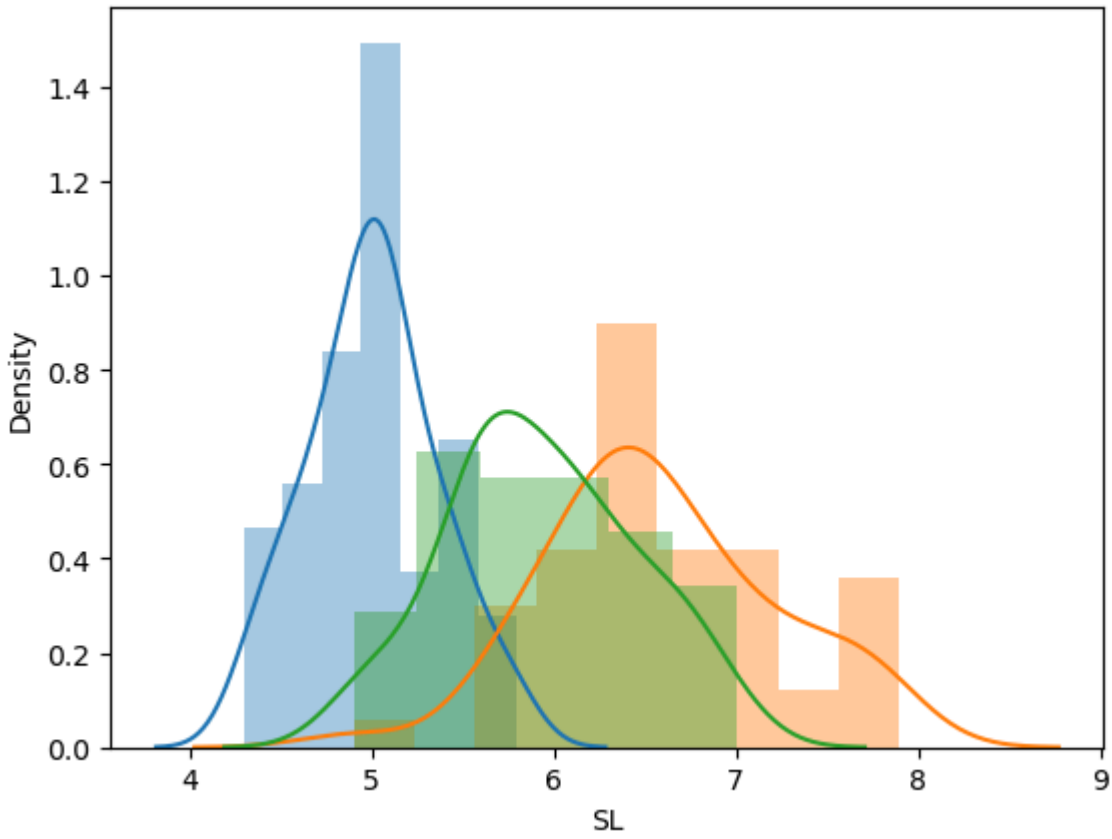
In [21]:
```
sns.kdeplot(iris.SL)
plt.show()
```



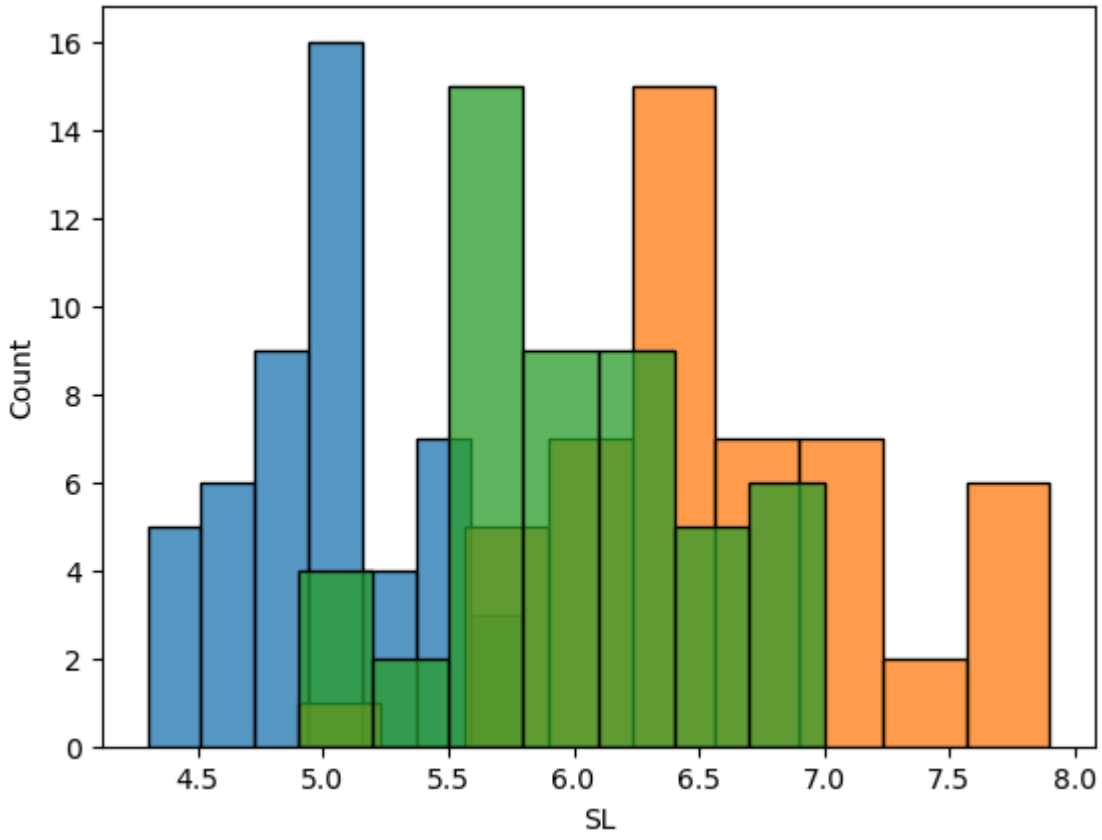
In [22]:
```
sns.kdeplot(iris.SL,fill=True,color='g',hatch='/')
plt.show()
```



In [23]:
```
sns.distplot(iris.SL[iris.Flower=='Iris-setosa'])
sns.distplot(iris.SL[iris.Flower=='Iris-virginica'])
sns.distplot(iris.SL[iris.Flower=='Iris-versicolor'])
plt.show()
```

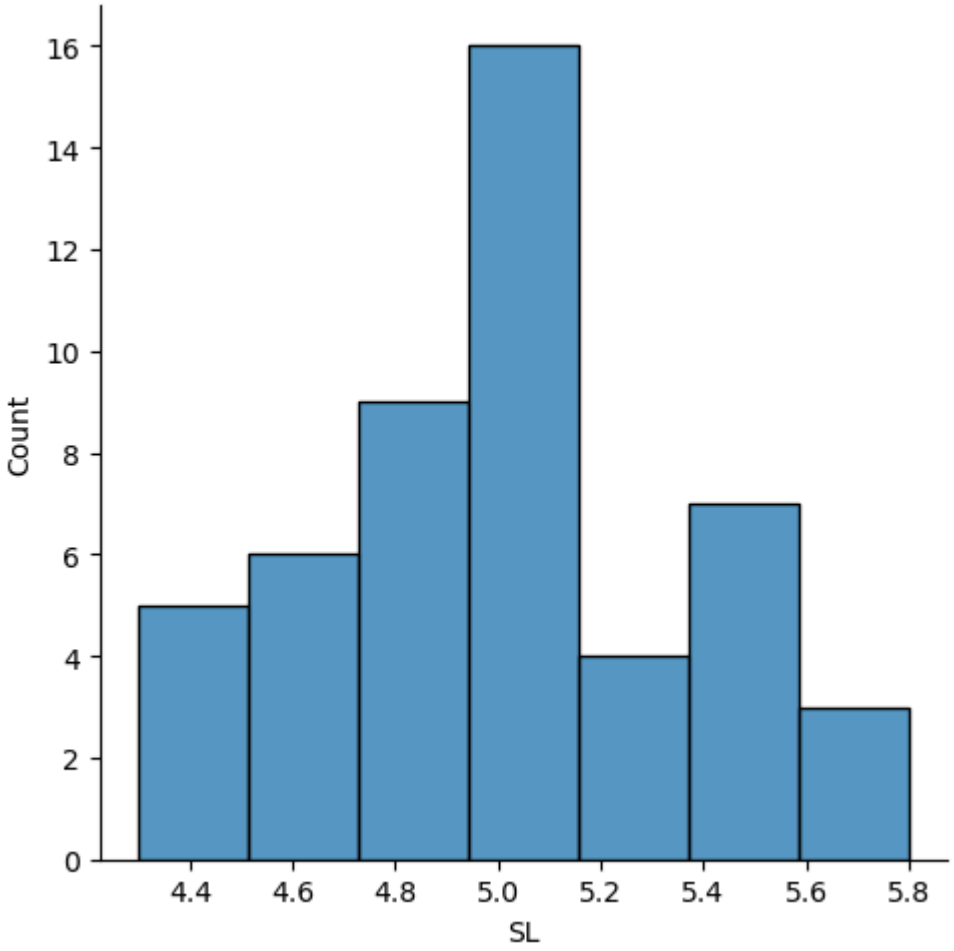


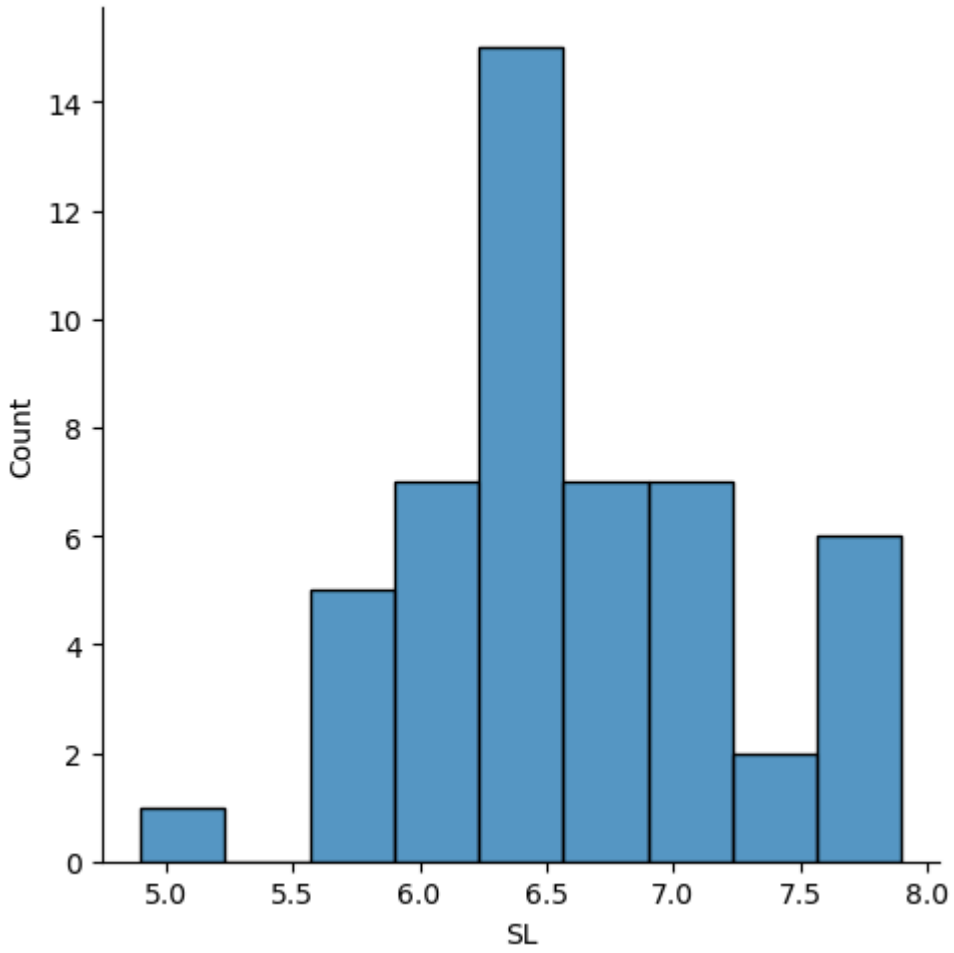
In [24]:
```
sns.histplot(iris.SL[iris.Flower=='Iris-setosa'])
sns.histplot(iris.SL[iris.Flower=='Iris-virginica'])
sns.histplot(iris.SL[iris.Flower=='Iris-versicolor'])
plt.show()
```

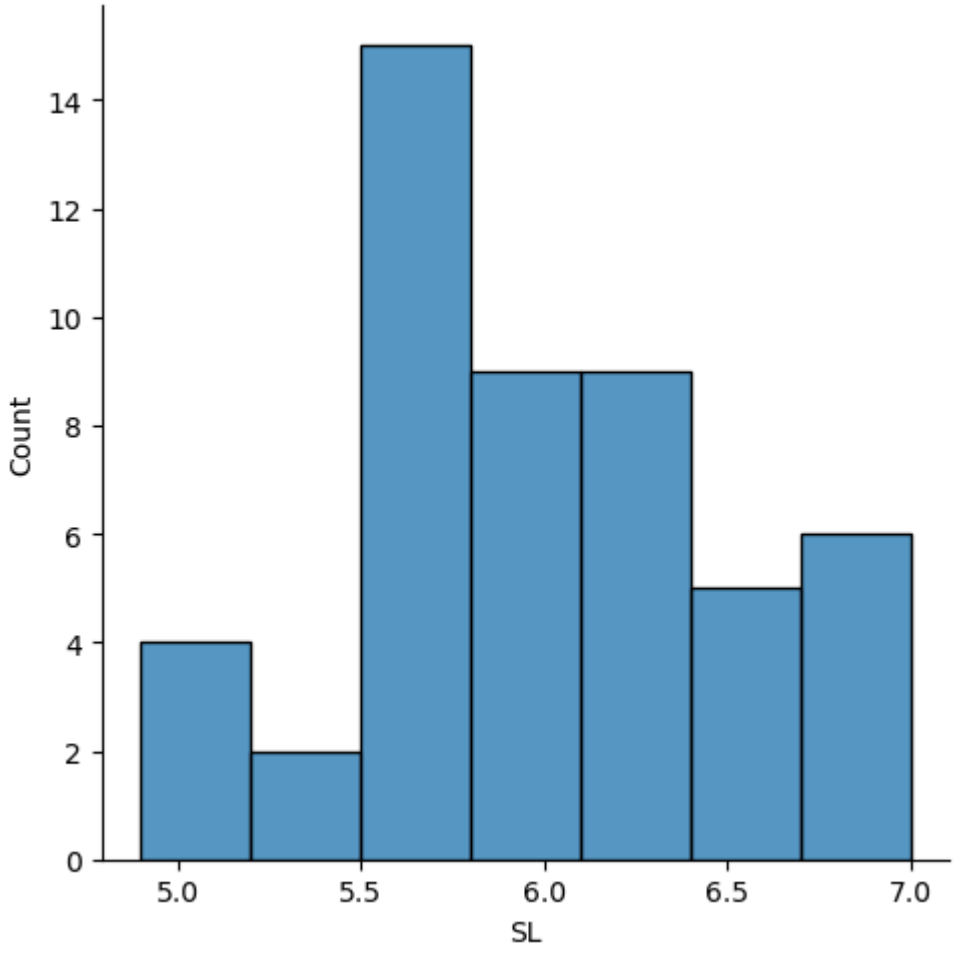


In [25]:
```
sns.displot(iris.SL[iris.Flower=='Iris-setosa'])
sns.displot(iris.SL[iris.Flower=='Iris-virginica'])
```

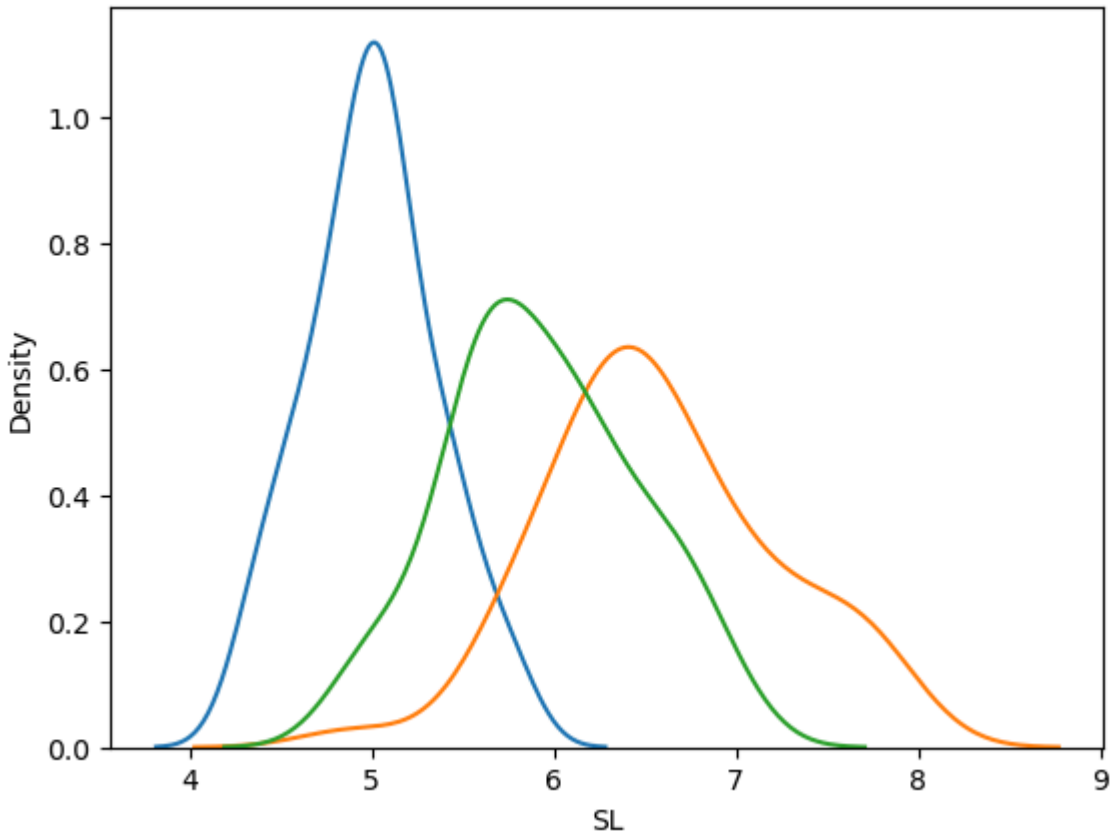
```
sns.displot(iris.SL[iris.Flower=='Iris-versicolor'])
plt.show()
```



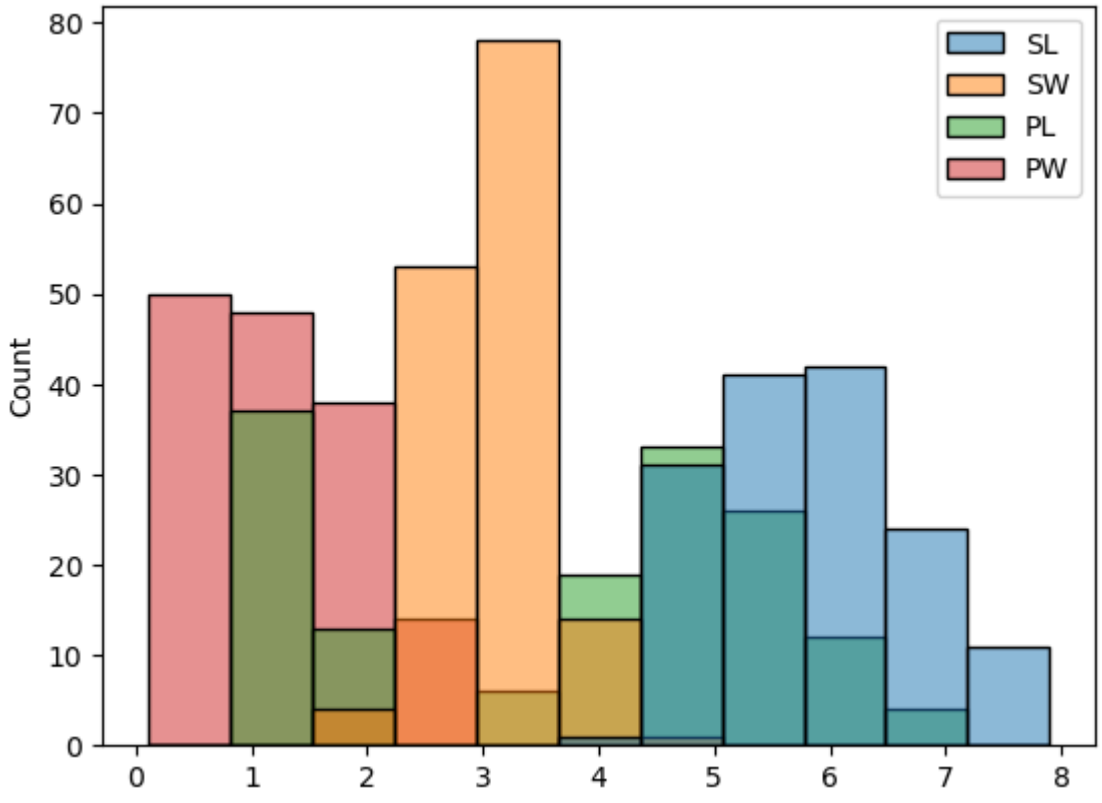

Count
SL
0
2
4
6
8
10
12
14
5.0
5.5
6.0
6.5
7.0
7.5
8.0


Count
SL
0
2
4
6
8
10
12
14
5.0
5.5
6.0
6.5
7.0

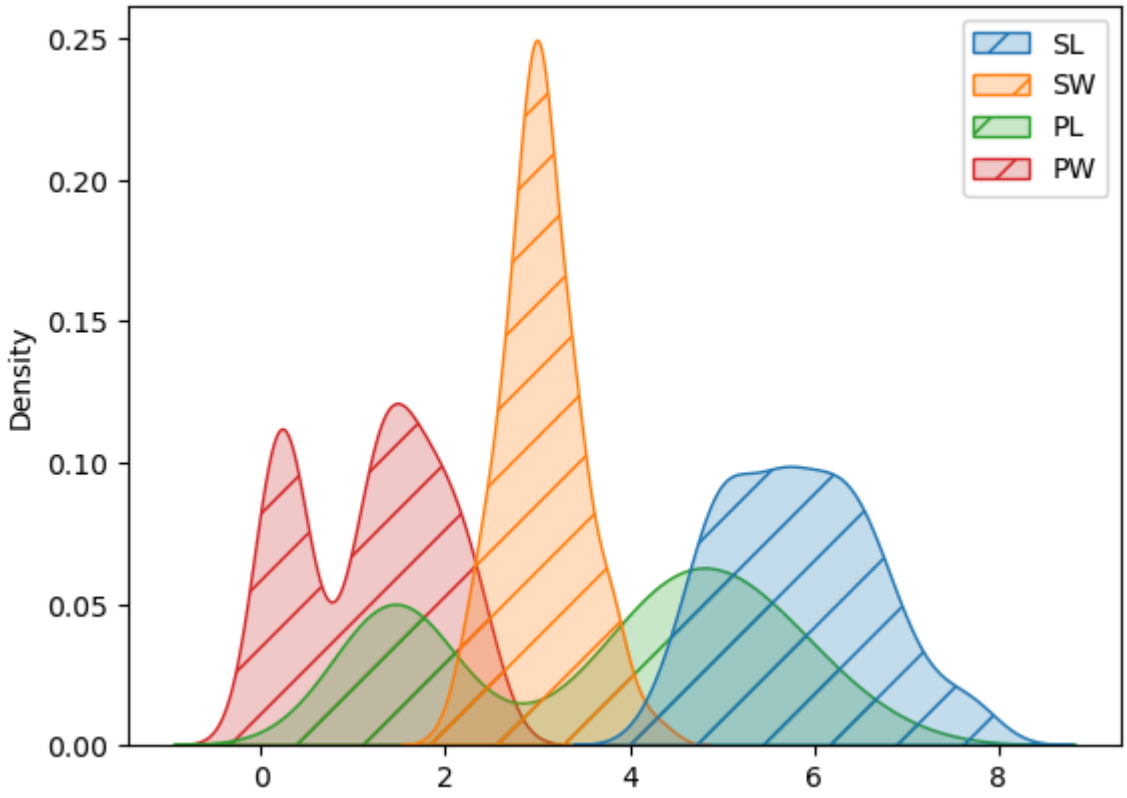
In [26]:
```
sns.kdeplot(iris.SL[iris.Flower=='Iris-setosa'])
sns.kdeplot(iris.SL[iris.Flower=='Iris-virginica'])
sns.kdeplot(iris.SL[iris.Flower=='Iris-versicolor'])
plt.show()
```



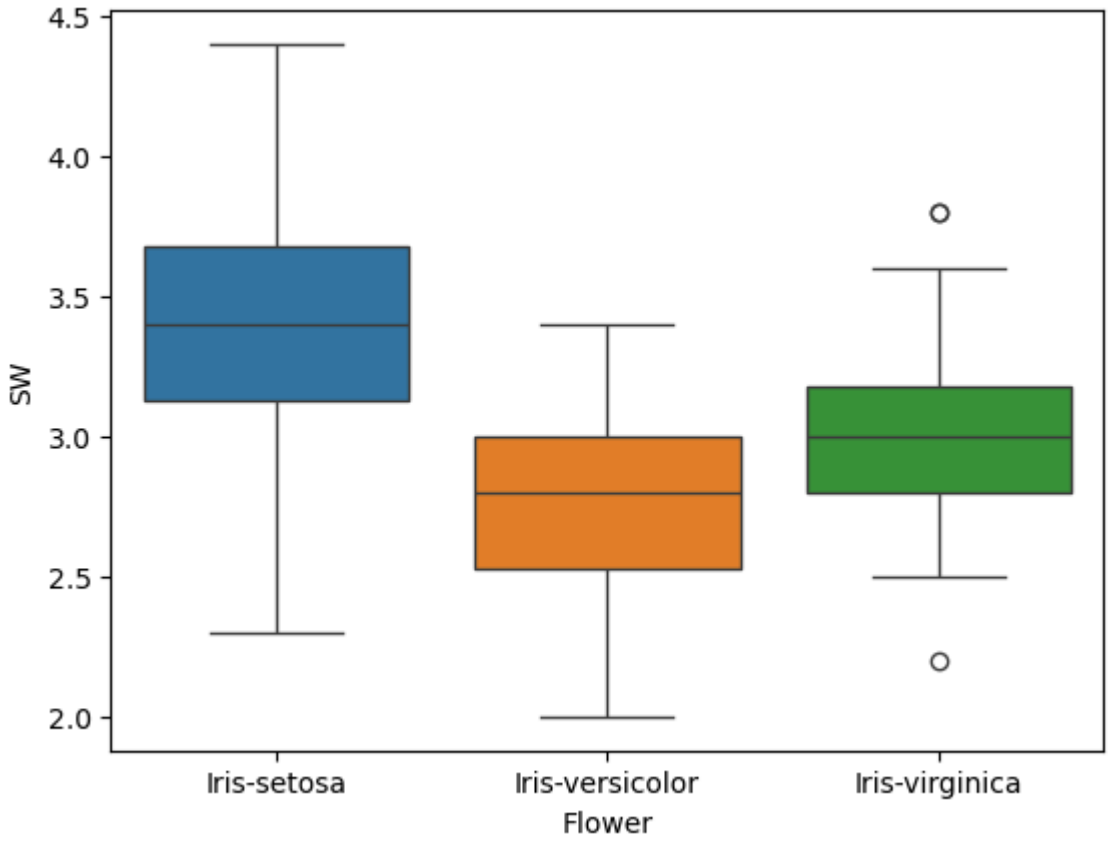
In [27]:
```
sns.histplot(data=iris)
plt.show()
```



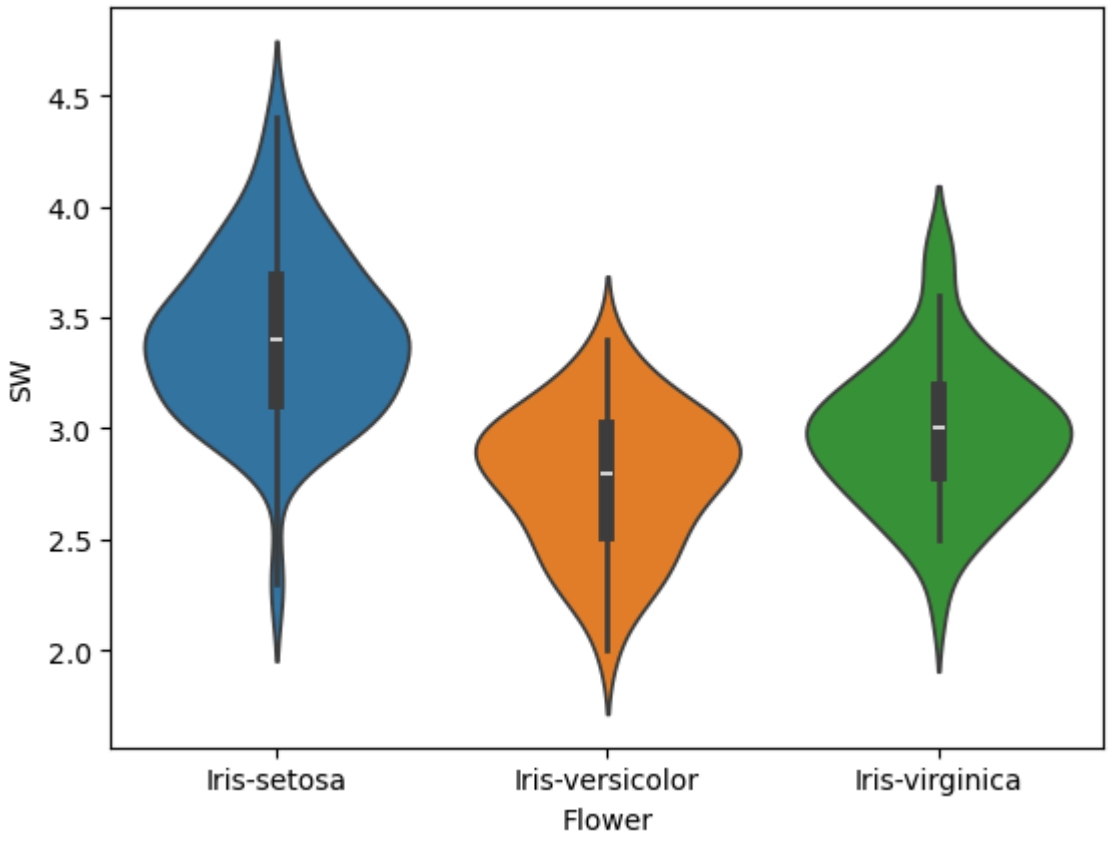
In [28]:
```
sns.kdeplot(data=iris,fill=True,hatch='/')
plt.show()
```



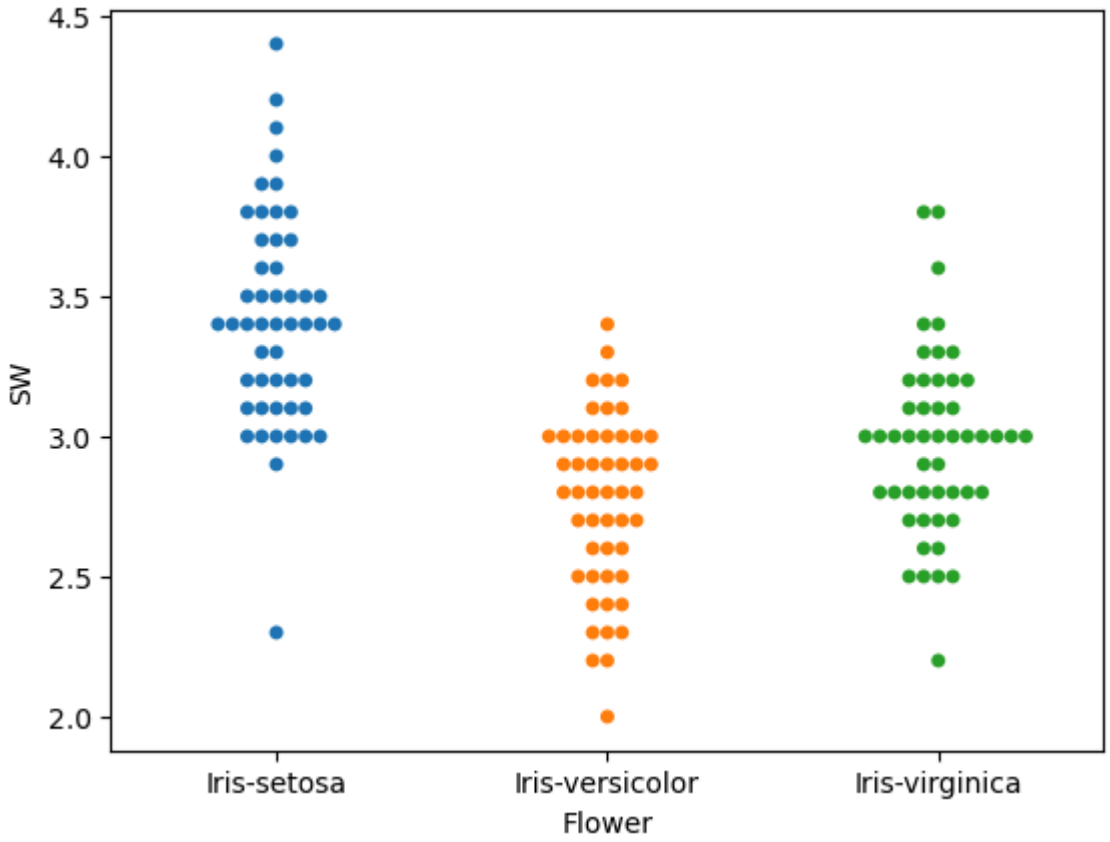
In [62]:
```
#Box Plot
sns.boxplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



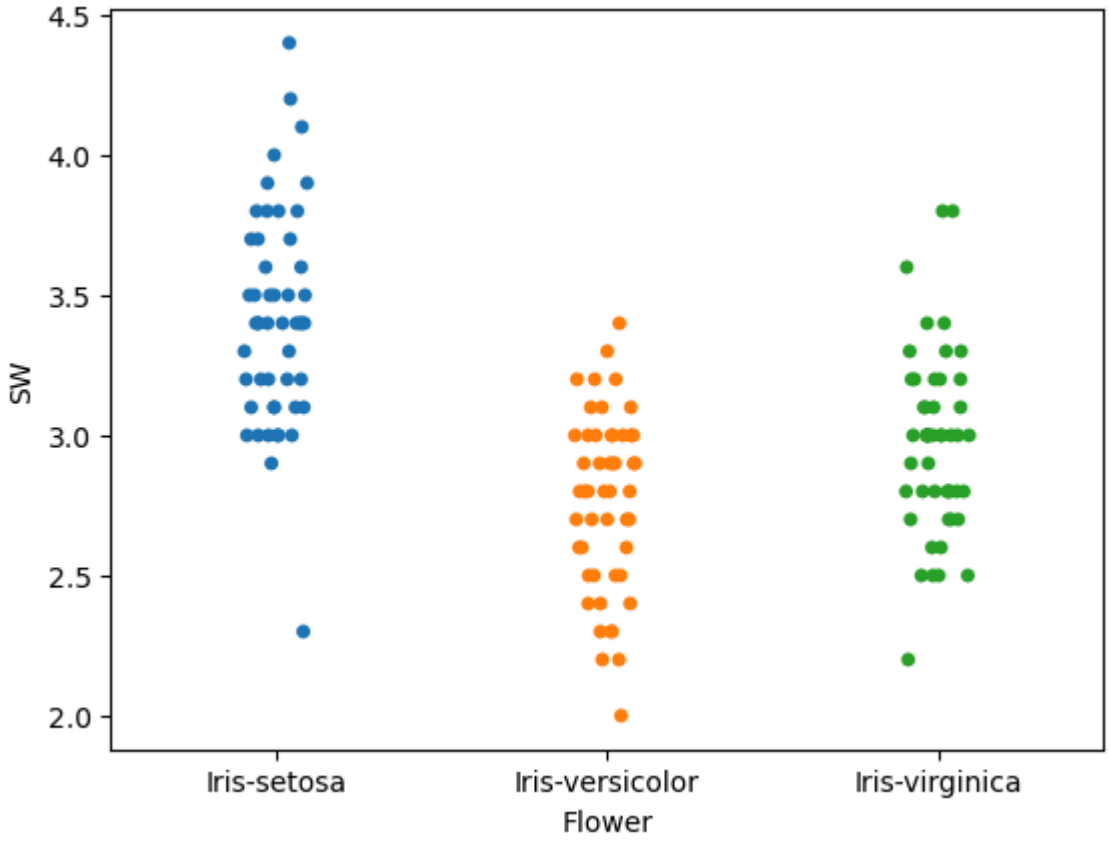
In [60]:
```
#Violin Plot
sns.violinplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



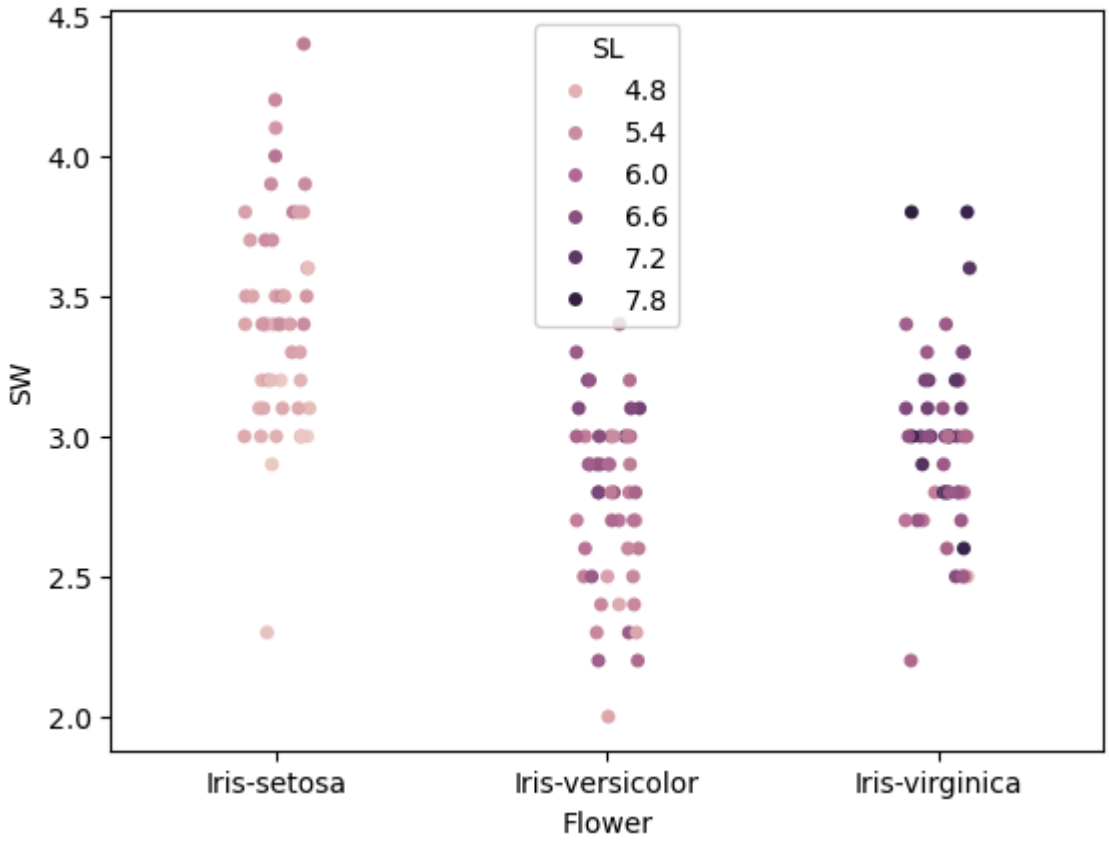
In [31]:
```
#Swarm Plot
sns.swarmplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



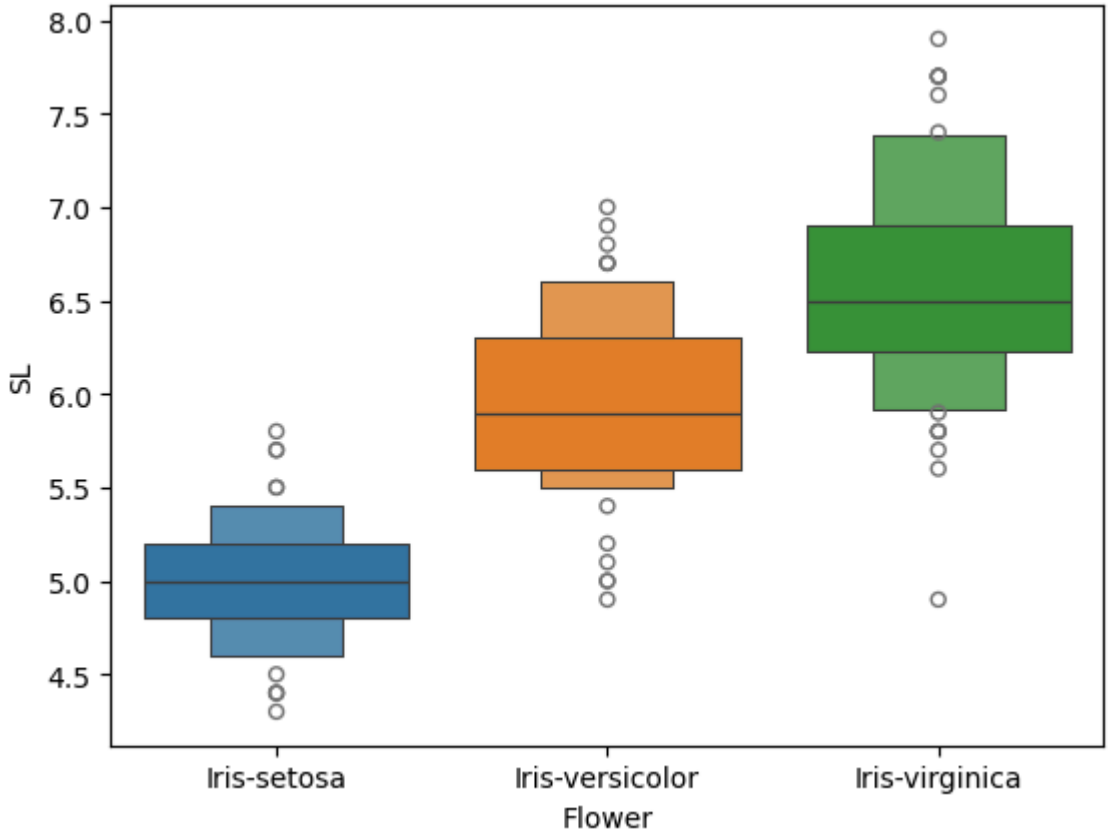
In [32]:
```
#Strip Plot
sns.stripplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



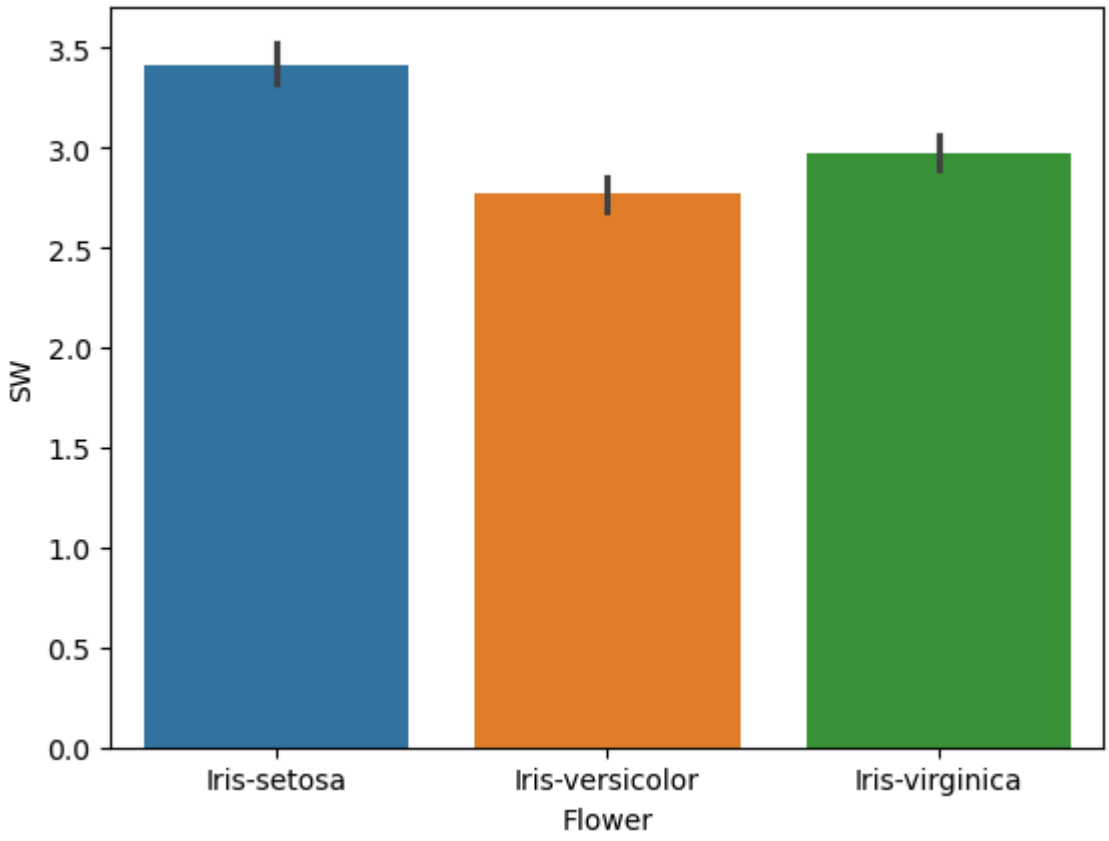
In [33]:
```
sns.stripplot(x=iris.Flower,y=iris.SW,hue=iris.SL)
plt.show()
```



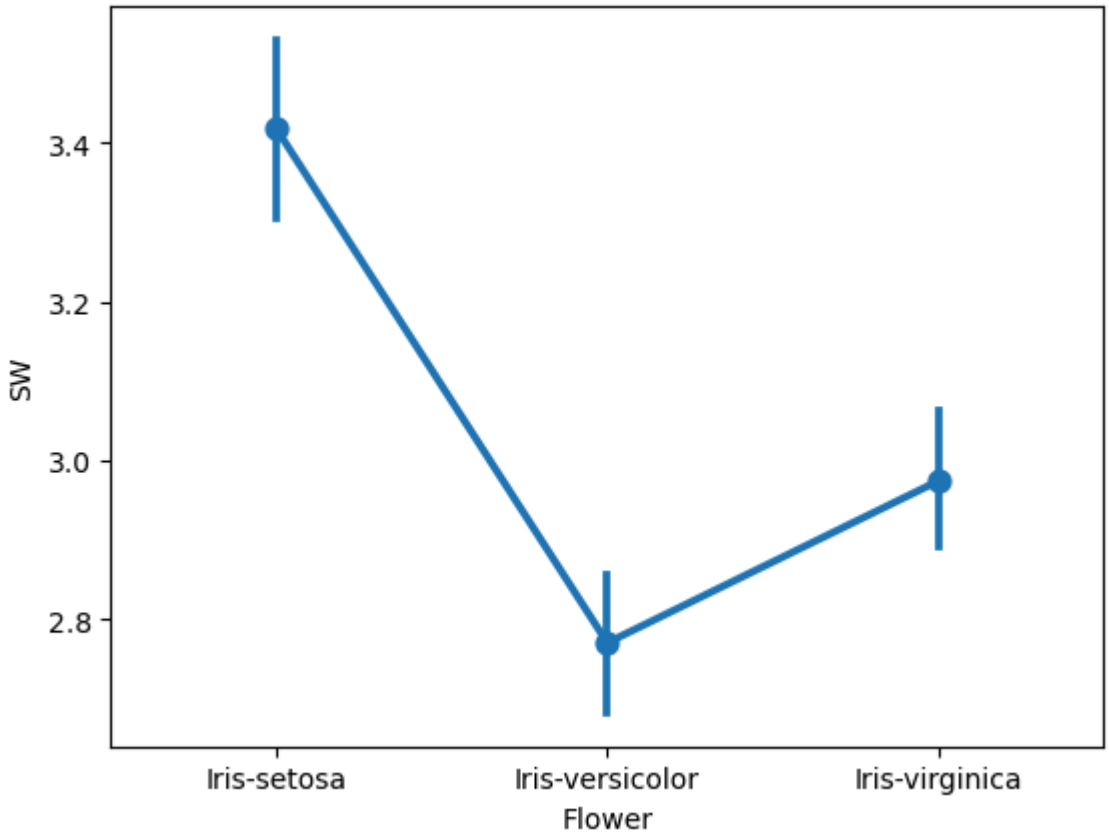
In [58]:
```
#Boxen Plot
sns.boxenplot(x=iris.Flower,y=iris.SL,hue=iris.Flower)
plt.show()
```



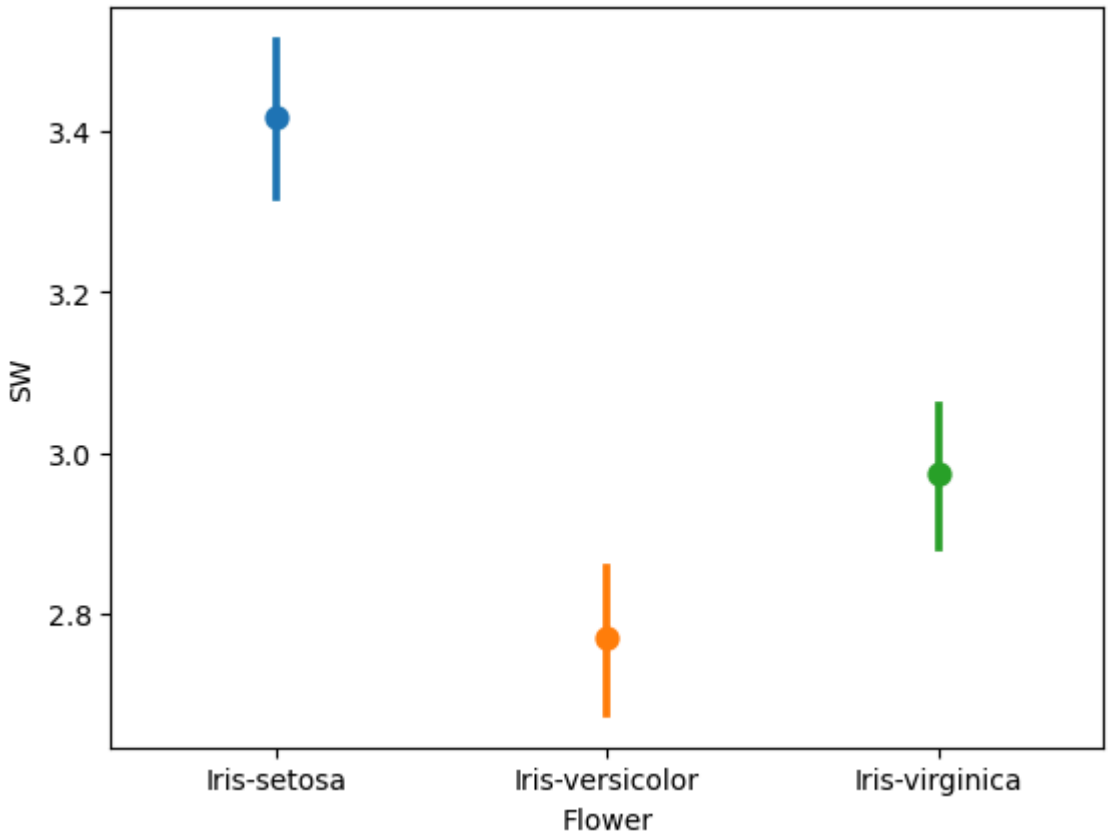
In [63]:
```
#Bar Plot
sns.barplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



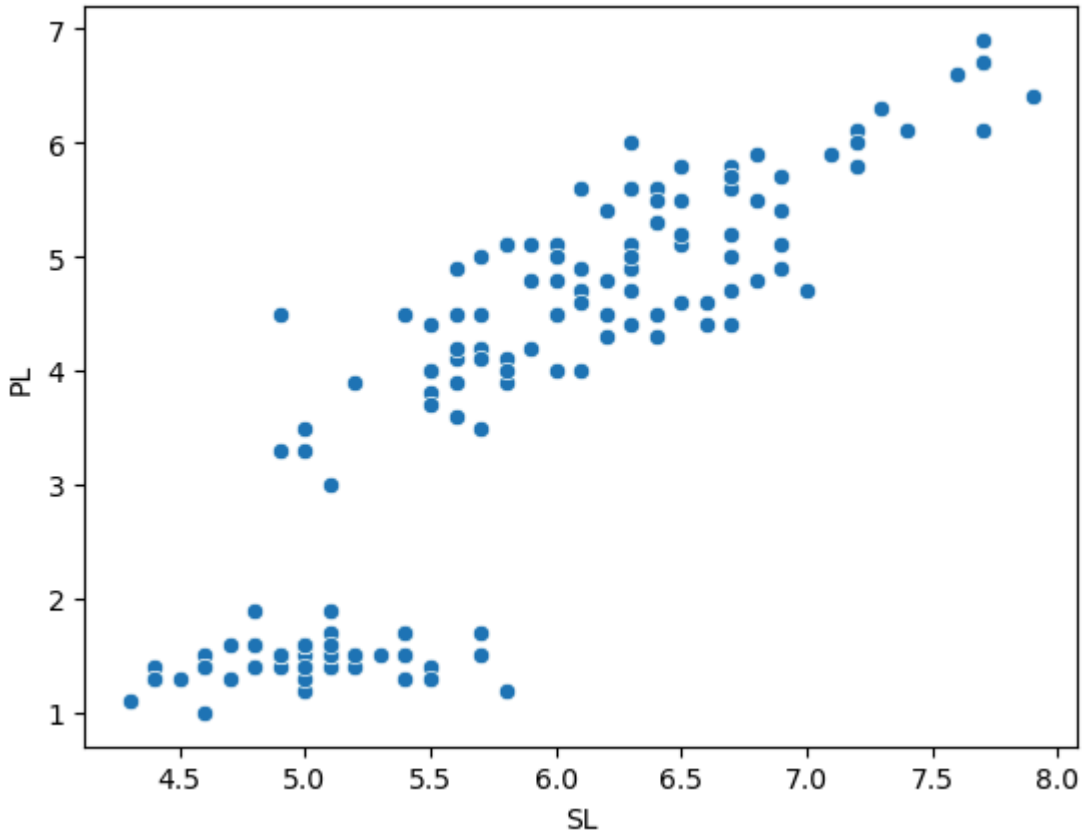
In [36]:
```
#Point Plot
sns.pointplot(x=iris.Flower,y=iris.SW)
plt.show()
```



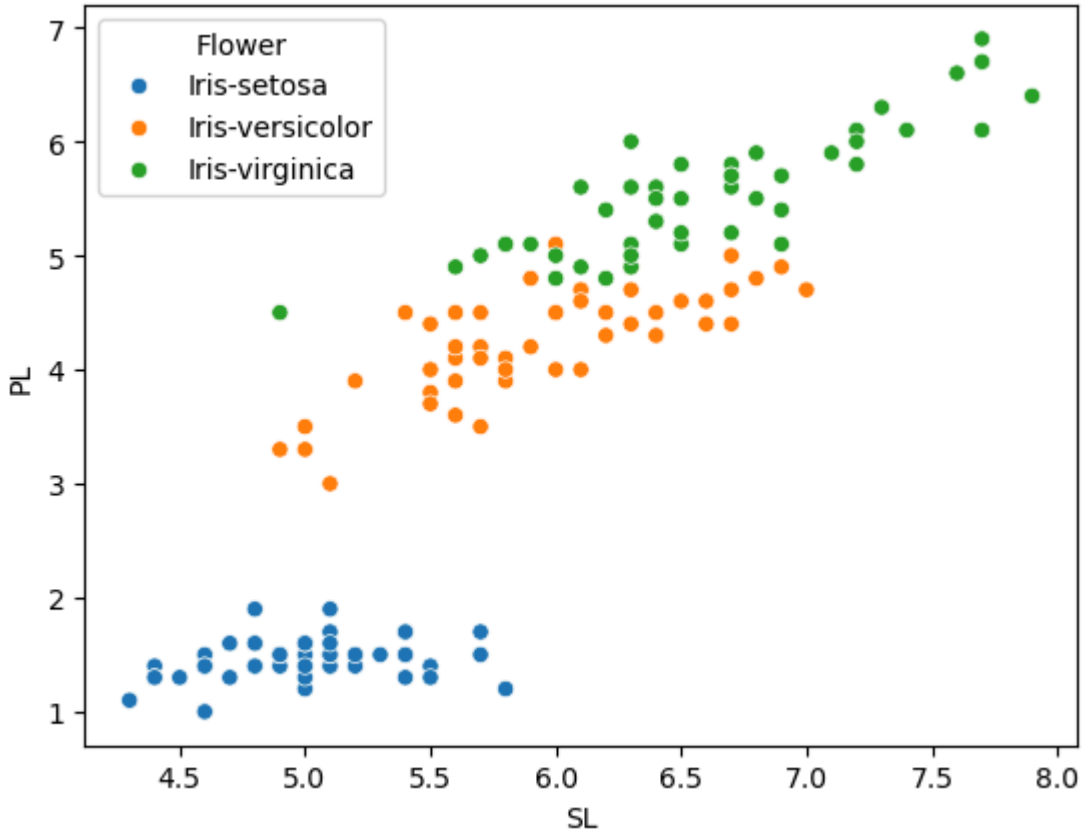
In [37]:
```
sns.pointplot(x=iris.Flower,y=iris.SW,hue=iris.Flower)
plt.show()
```



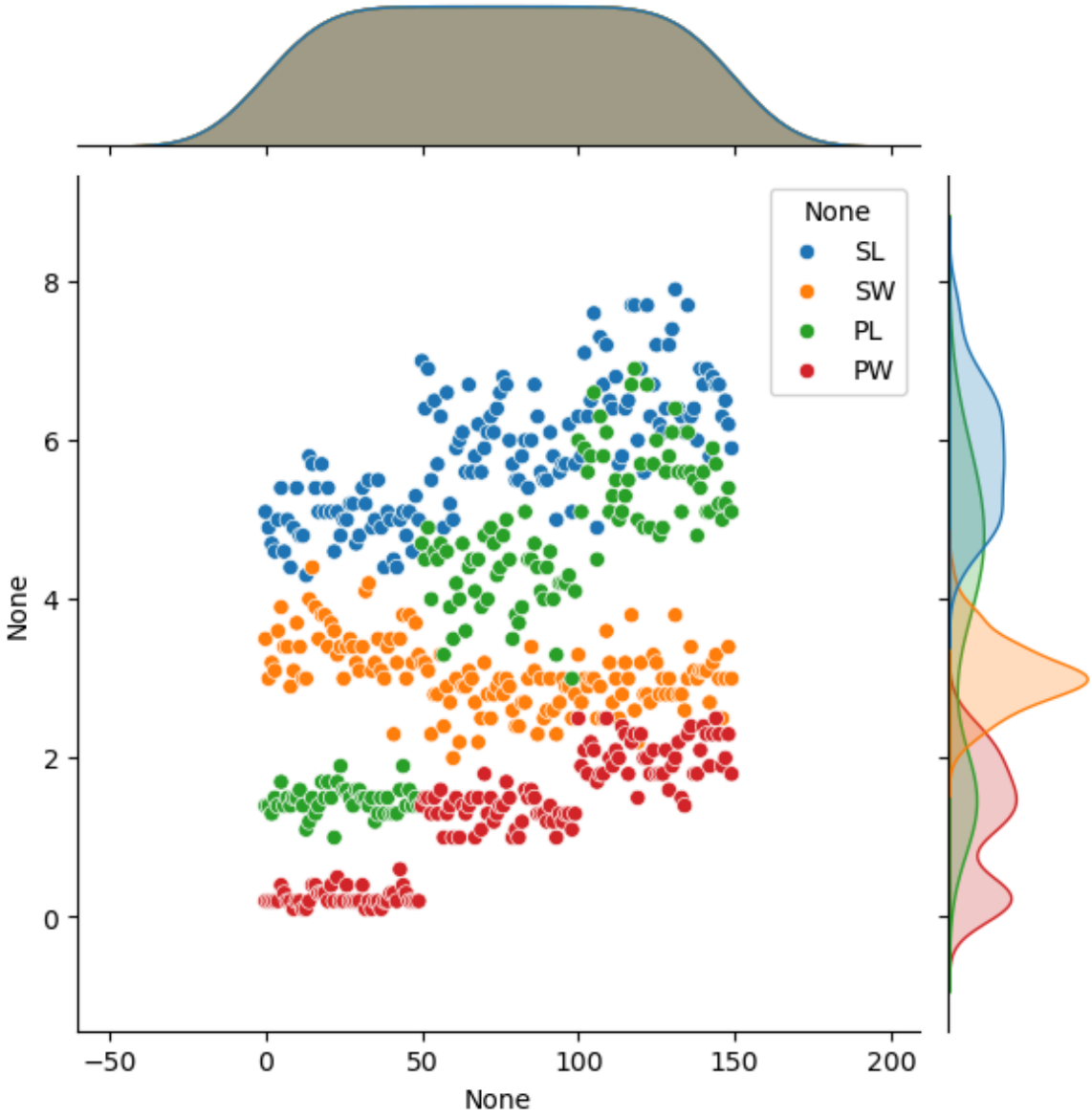
In [38]:
```
#Scatter Plot
sns.scatterplot(x=iris.SL,y=iris.PL)
plt.show()
```



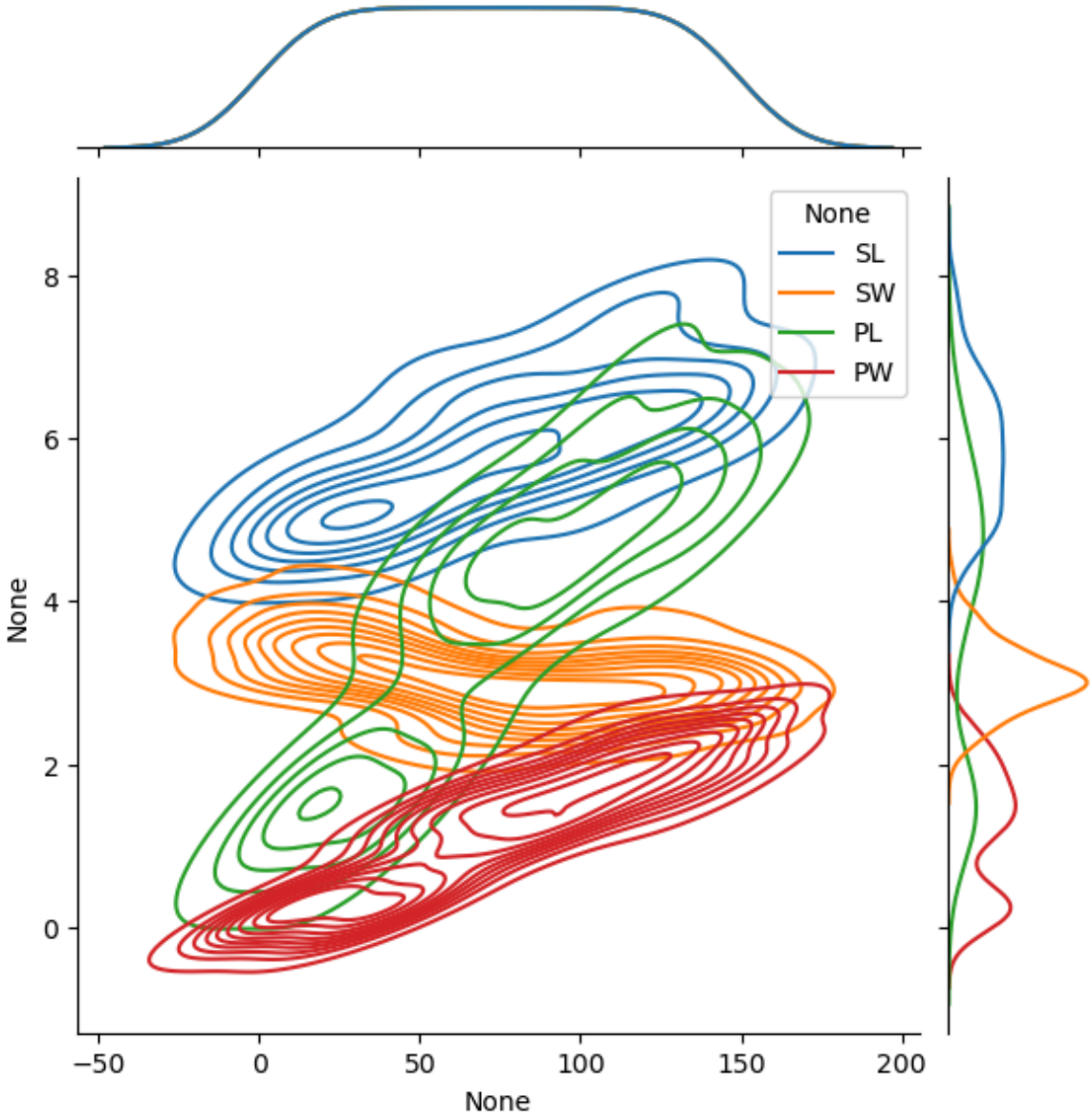
In [39]:
```
sns.scatterplot(x=iris.SL,y=iris.PL,hue=iris.Flower)
plt.show()
```



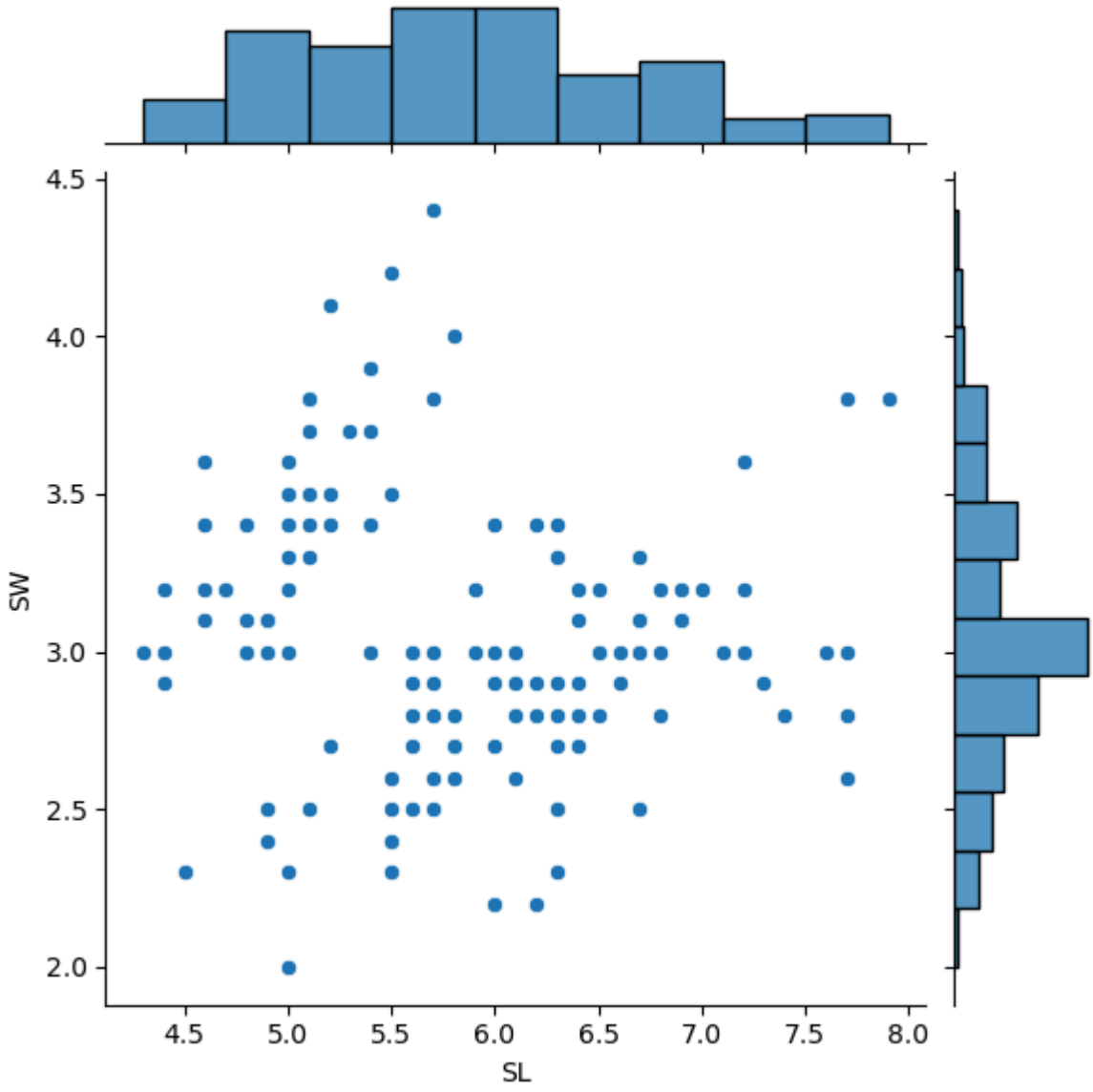
In [40]:
```
#Joint plot
sns.jointplot(data=iris)
plt.show()
```



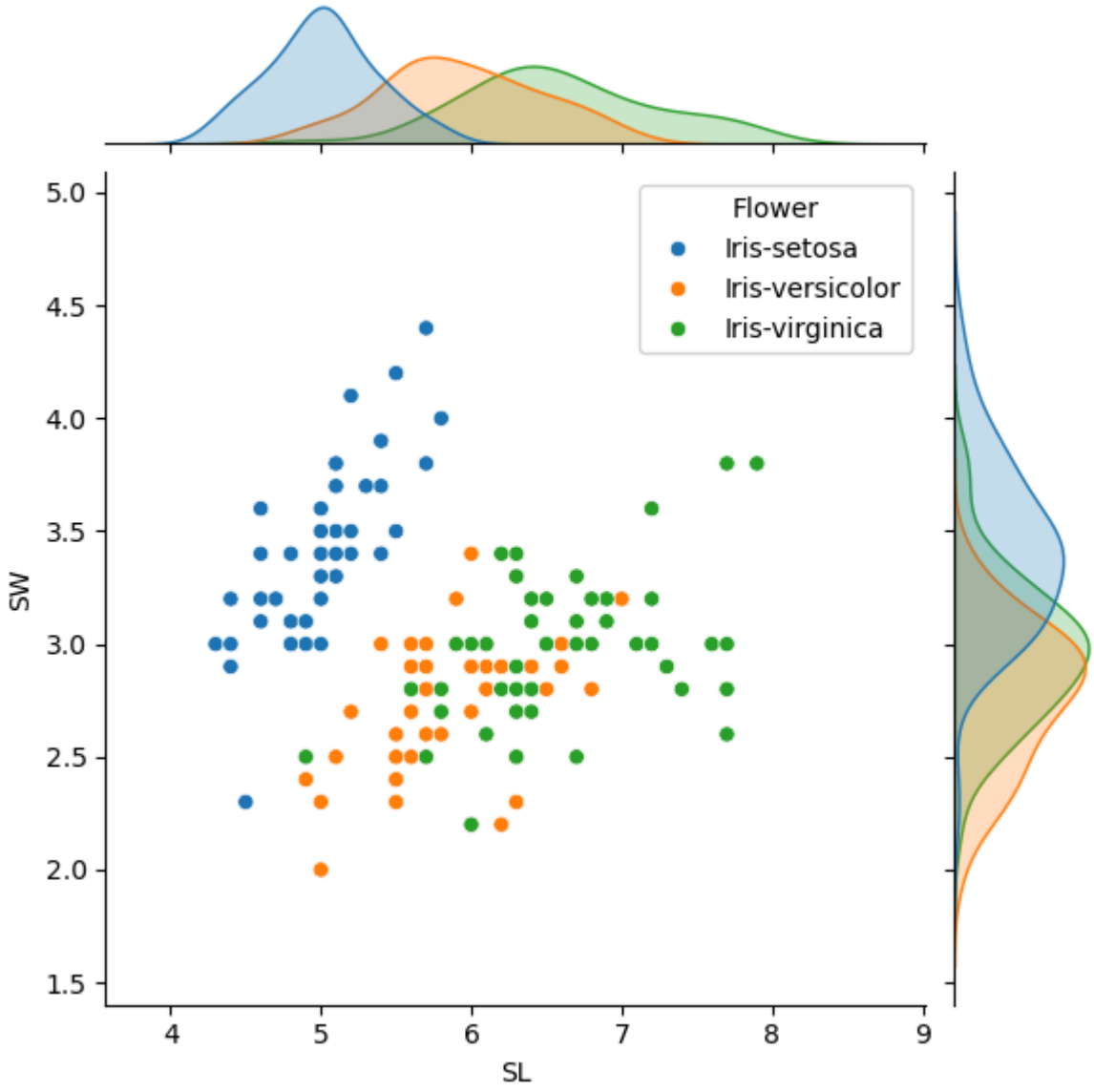
In [41]:
```
sns.jointplot(data=iris,kind='kde')
plt.show()
```



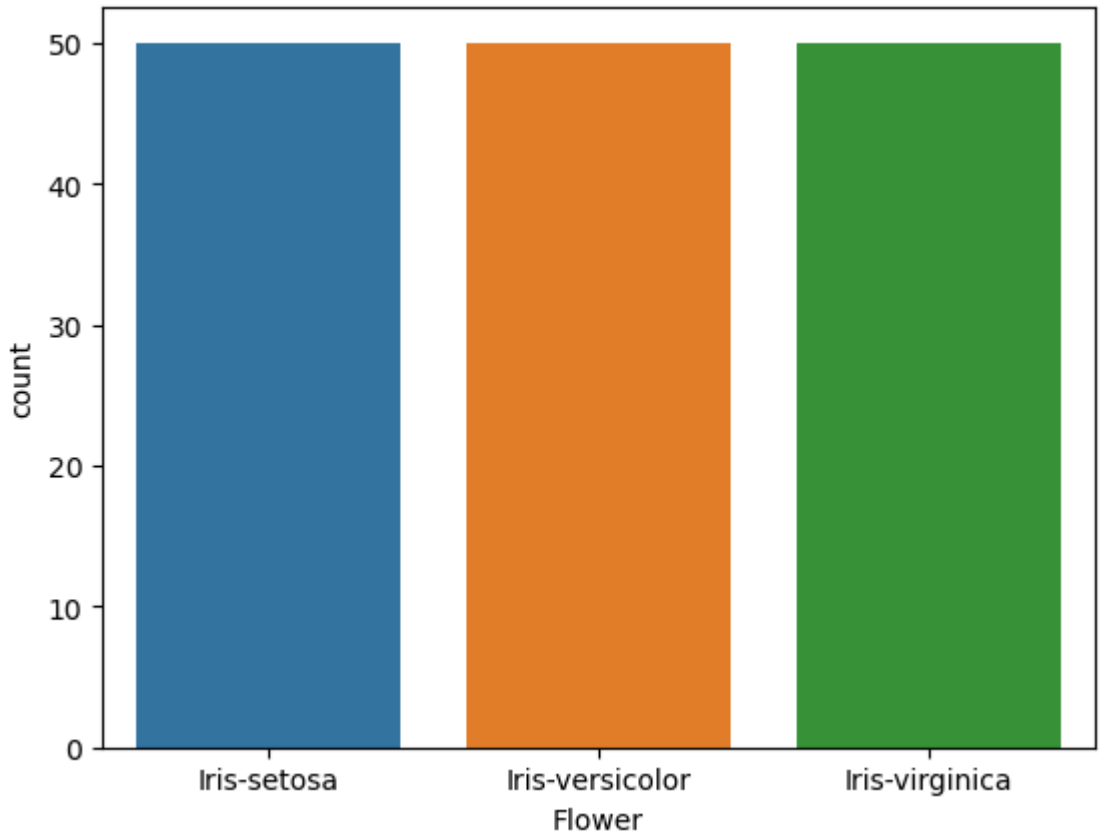
In [42]:
```
sns.jointplot(x=iris.SL,y=iris.SW)
plt.show()
```



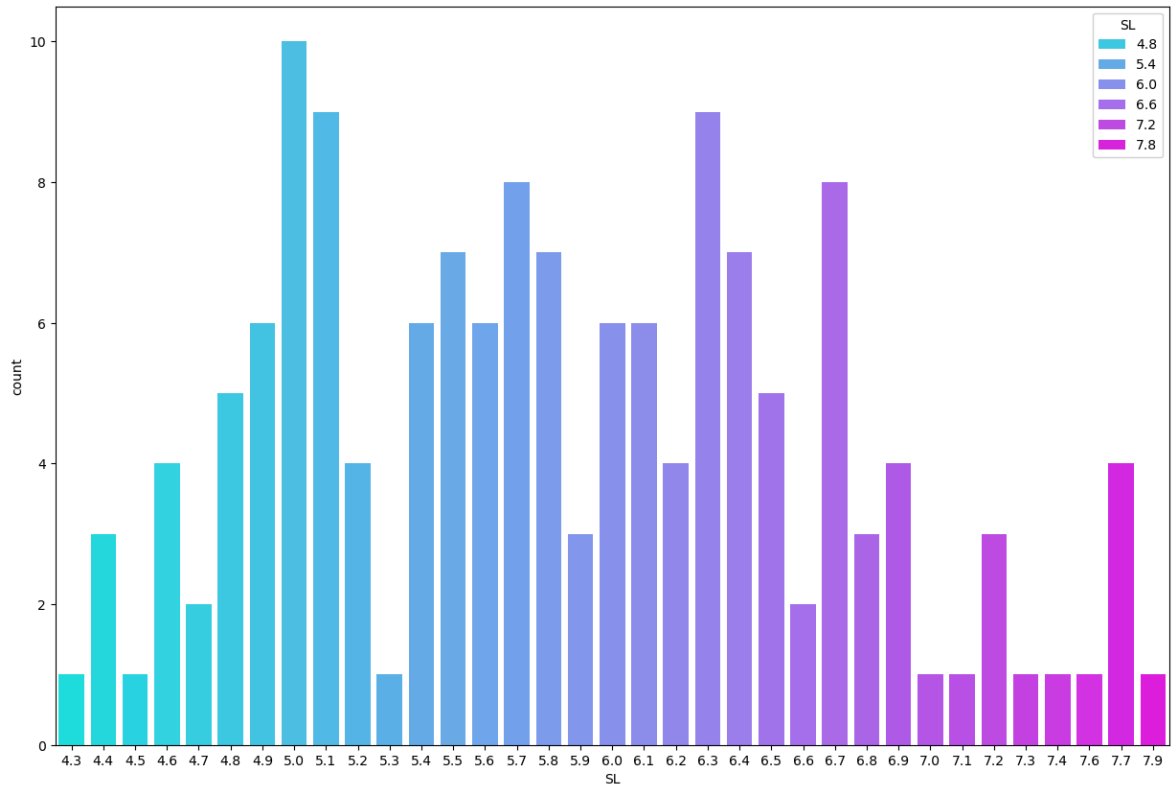
In [43]:
```
sns.jointplot(x=iris.SL,y=iris.SW,hue=iris.Flower)
plt.show()
```



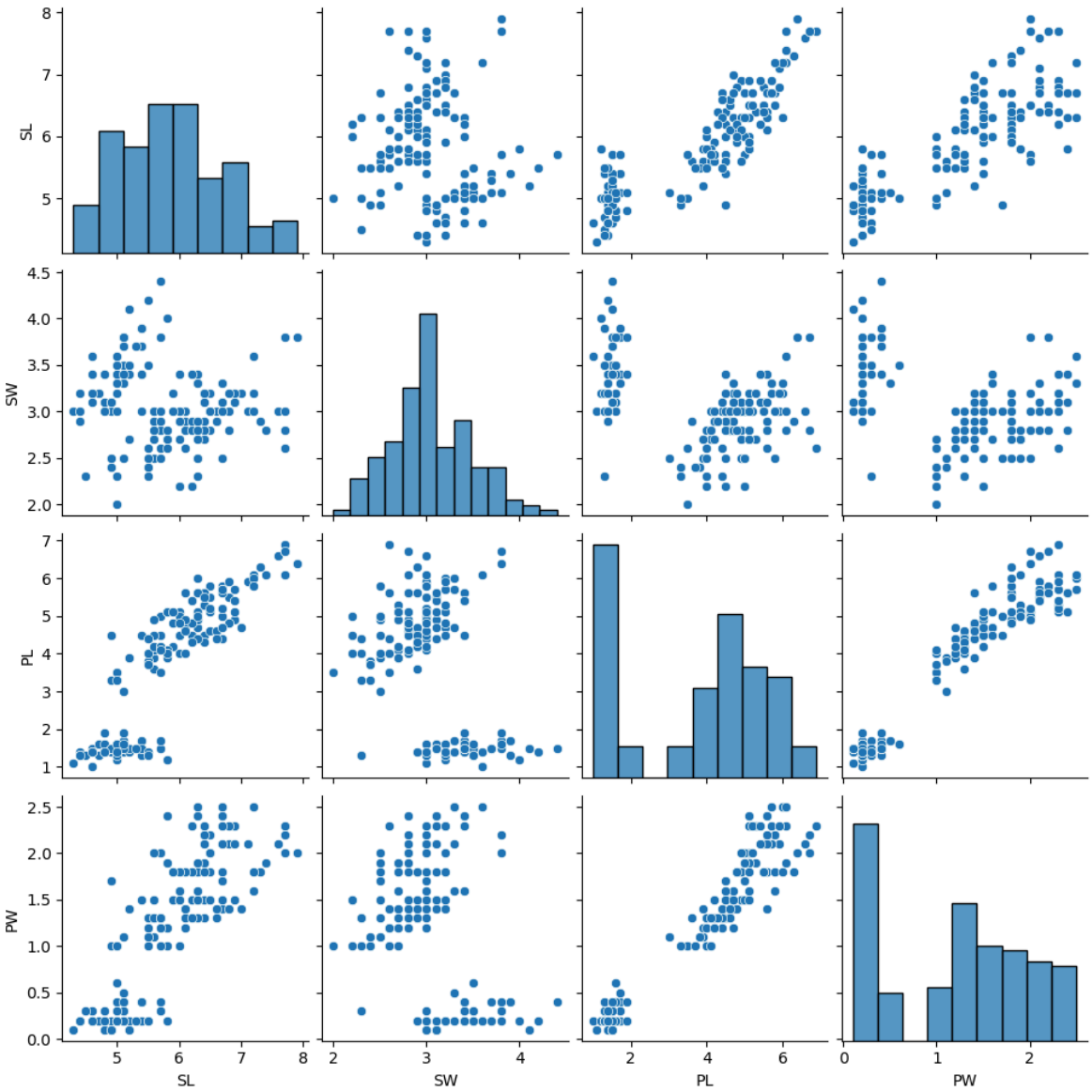
In [65]:
```
#Count Plot
sns.countplot(x=iris.Flower,hue=iris.Flower)
plt.show()
```



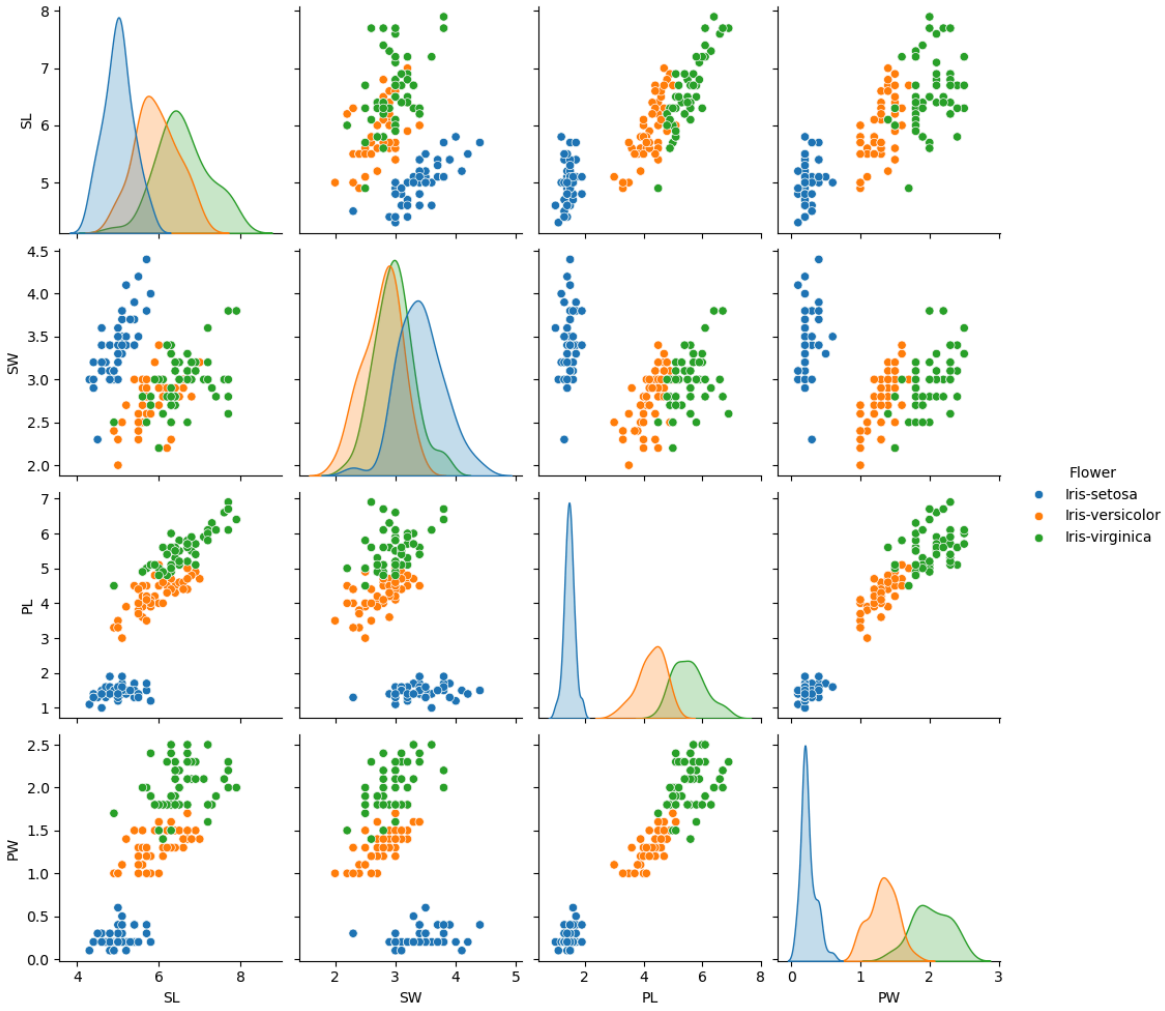
In [68]:
```
plt.figure(figsize=(15,10))
sns.countplot(x=iris.SL,hue=iris.SL,palette='cool')
plt.show()
```



In [46]:
```
#Pair Plot
sns.pairplot(iris)
plt.show()
```



In [47]:
```
sns.pairplot(iris,hue='Flower')
plt.show()
```



In [48]:
```
d = iris.drop('Flower',axis=1)
d
```

Out[48]:

	SL	SW	PL	PW
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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [49]:
```
#Heat Map
c = d.corr()
c
```

Out[49]:

	SL	SW	PL	PW
SL	1.000000	-0.109369	0.871754	0.817954
SW	-0.109369	1.000000	-0.420516	-0.356544
PL	0.871754	-0.420516	1.000000	0.962757
PW	0.817954	-0.356544	0.962757	1.000000

In [50]:
```
!pip install seaborn --upgrade
```

```
Requirement already satisfied: seaborn in e:\anaconda\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in e:\anaconda\lib\site-packa
ges (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in e:\anaconda\lib\site-packages (from
seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in e:\anaconda\lib\site-pa
ckages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in e:\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in e:\anaconda\lib\site-packages (fro
m matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in e:\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in e:\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in e:\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in e:\anaconda\lib\site-packages (fr
om matplotlib!=3.6.1,>=3.4->seaborn) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in e:\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in e:\anaconda\lib\site-packa
ges (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in e:\anaconda\lib\site-packages (fro
m pandas>=1.2->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in e:\anaconda\lib\site-packages (f
rom pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in e:\anaconda\lib\site-packages (from py
thon-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
```

In [51]:
```
sns.heatmap(c,annot=True,cmap='Blues')
plt.show()
```


SL
SW
PL
PW
1
-0.11
0.87
0.82
-0.11
1
-0.42
-0.36
0.87
-0.42
1
0.96
0.82
-0.36
0.96
1
SL
SW
PL
PW
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

In [ ]:

In [ ]: