Name:- Anushka Pradeep Jadhav

Roll No.:- 23091015

Q1. Data Visualization and Statistical Measures:

For this question, you are required to analyse the iris dataset (iris.csv) using Python. Perform all possible data visualization techniques (histograms, scatter plots, box plots, etc.) on all numerical columns of the dataset. Additionally, calculate all possible statistical measures (mean, median, mode, standard deviation, etc.) for each numerical column.

In [1]:
```
import pandas as pd #Importing Panda Library
```

In [2]:
```
df=pd.read_csv('Iris.csv') #Importing Dataset
```

In [3]:
```
df #Displaying Dataset
```

Out[3]:

	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

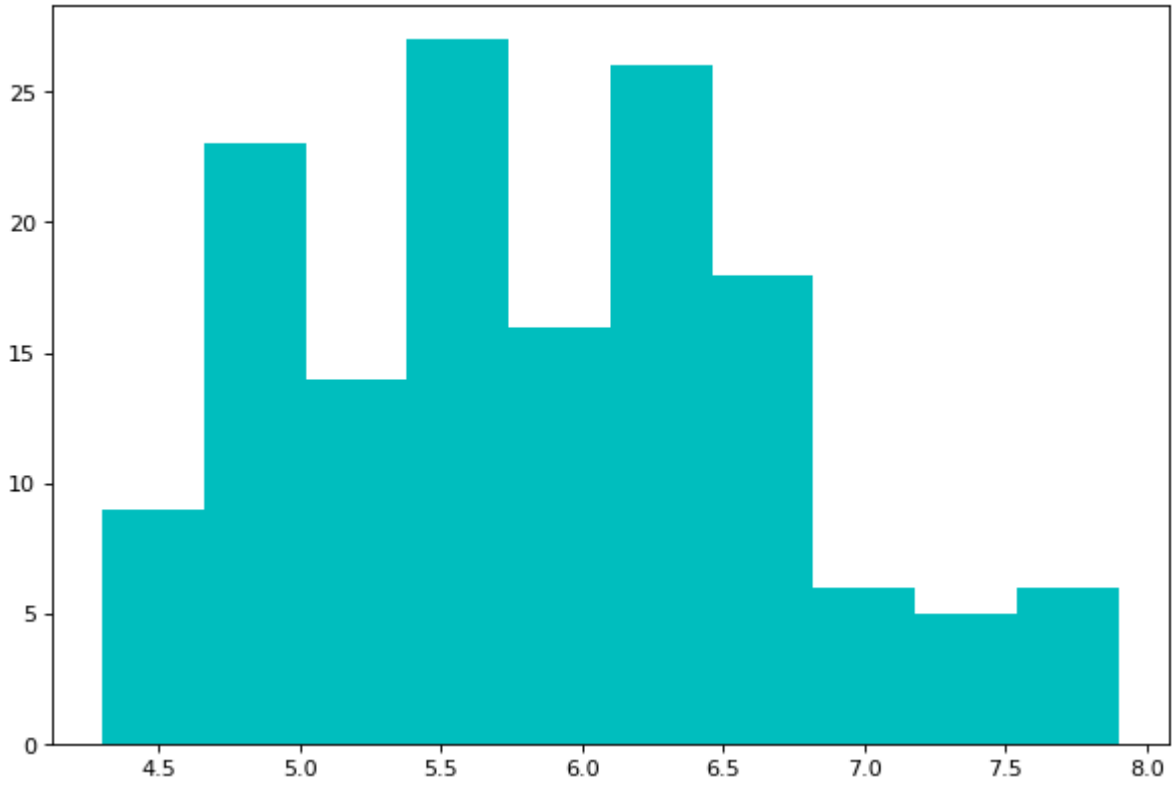
In [4]:
```
#Importing Library for Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Figure Properties
# rcParams : Runtime Configuration Parameters
plt.rcParams['figure.figsize']=(9,6)
plt.rcParams['figure.dpi']=80
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
#Performing all possible data visualization techniques on all numerical columns of the dataset.
#Here we plot all the plots of column SepalLengthCm in cyan colour, SepalWidthCm in Magenta colour,
#PetalLengthCm in Green colour, PetalWidthCm in red colour.
```

In [6]:
```
#Plotting Histogram for SepalLength, SepalWidth, PetalLength, PetalWidth
```

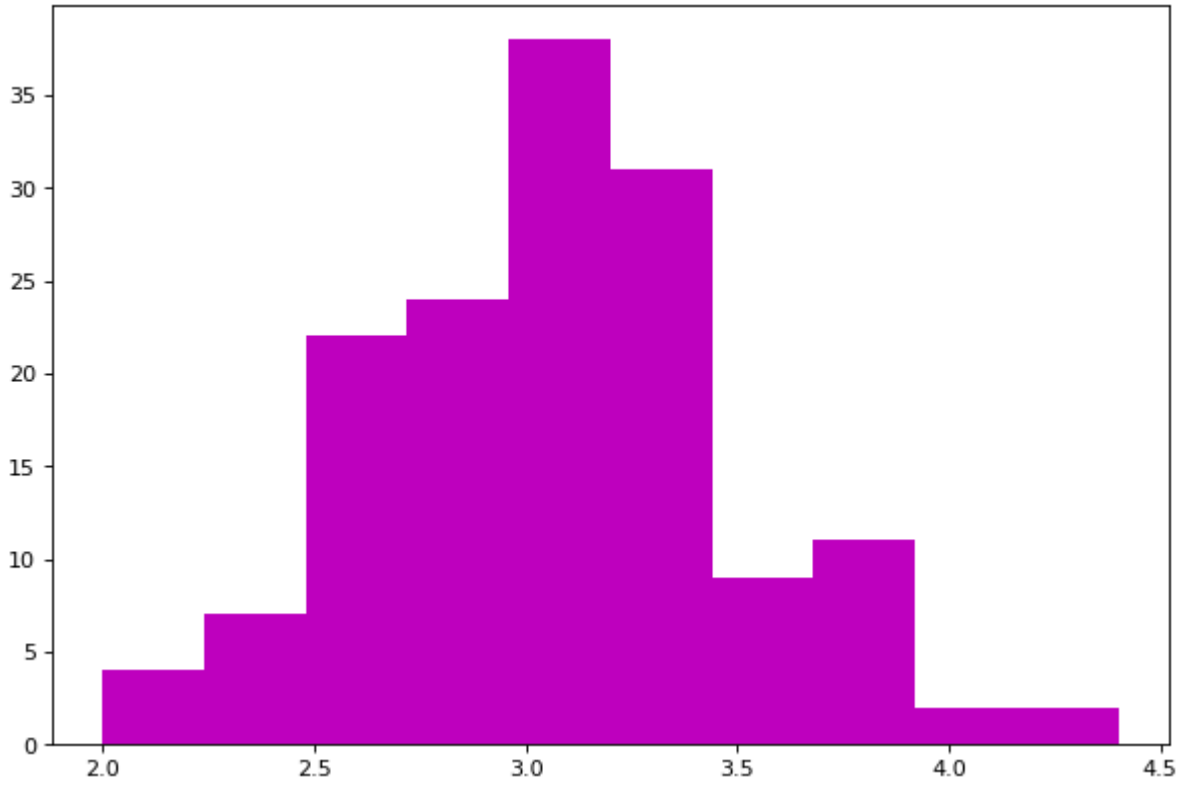
In [7]:
```
plt.hist(df['SepalLengthCm'],color='c')
```

Out[7]:
```
(array([ 9., 23., 14., 27., 16., 26., 18.,  6.,  5.,  6.]),
 array([4.3 , 4.66, 5.02, 5.38, 5.74, 6.1 , 6.46, 6.82, 7.18, 7.54, 7.9 ]),
 <BarContainer object of 10 artists>)
```



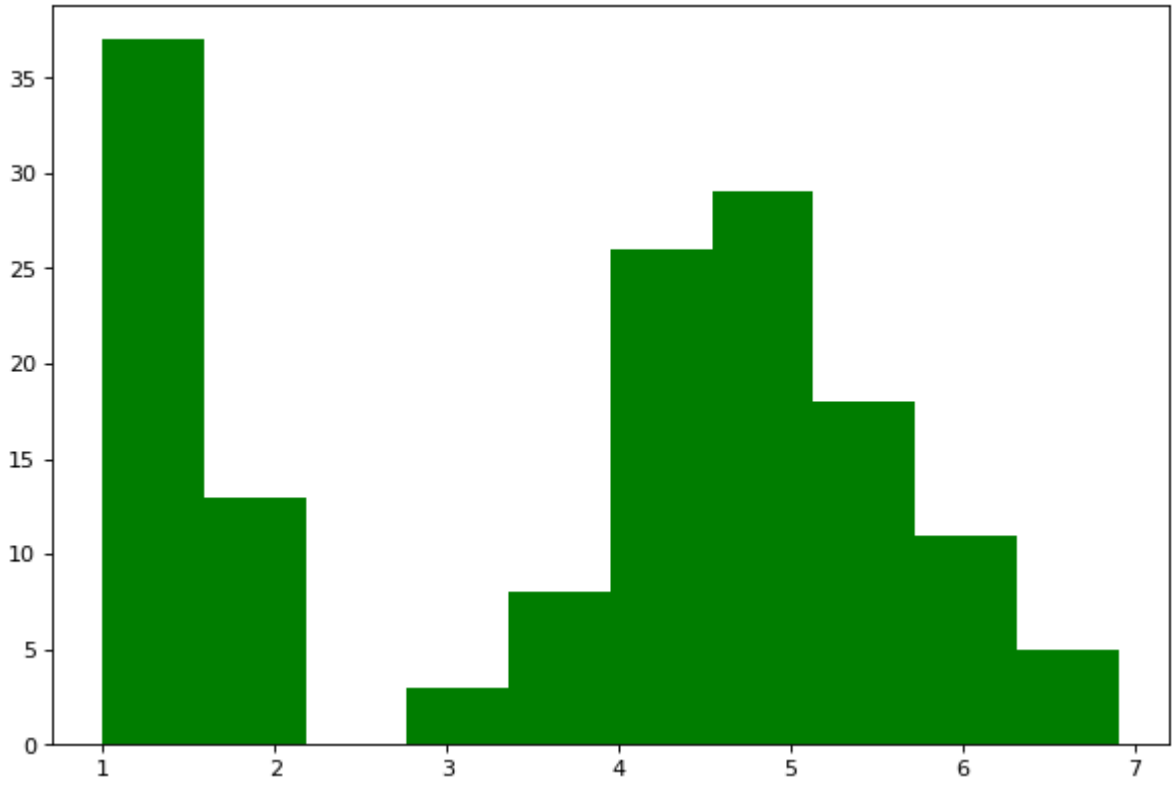
In [8]:
```
plt.hist(df['SepalWidthCm'],color='m')
```

Out[8]:
```
(array([ 4.,  7., 22., 24., 38., 31.,  9., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```



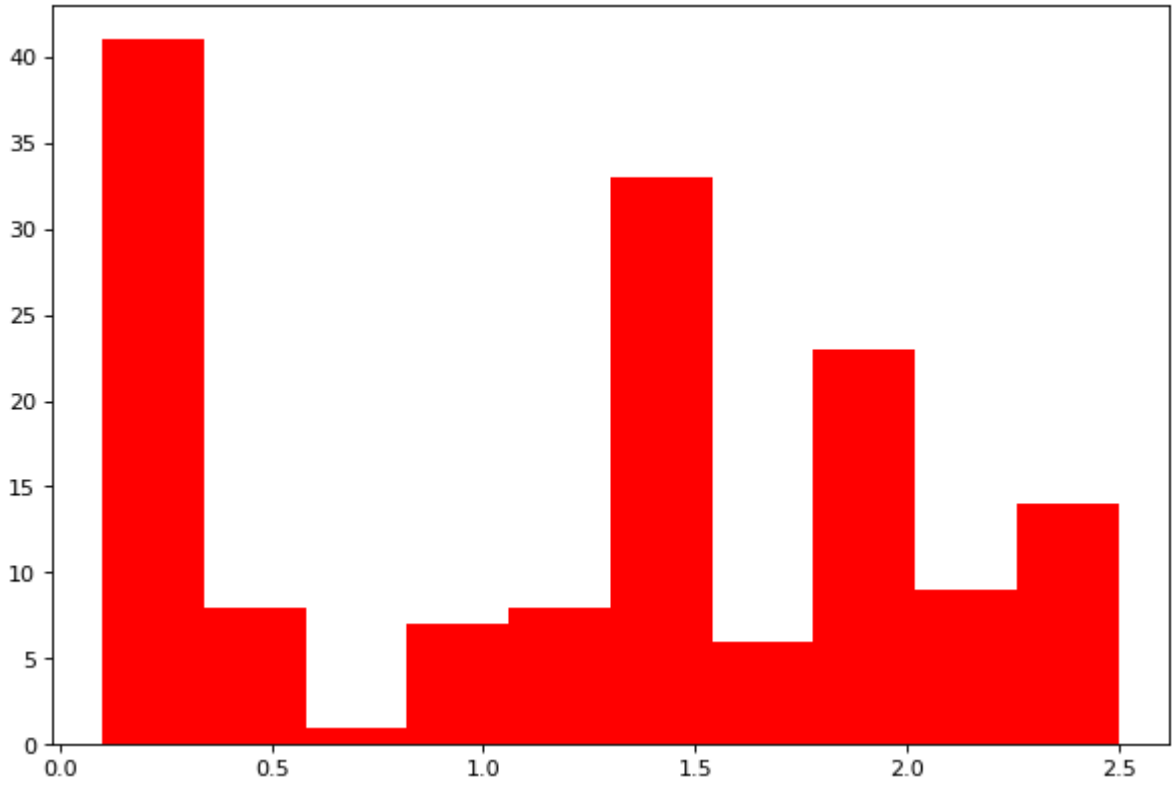
In [9]:
```
plt.hist(df['PetalLengthCm'],color='g')
```

Out[9]:
```
(array([37., 13.,  0.,  3.,  8., 26., 29., 18., 11.,  5.]),
 array([1.  , 1.59, 2.18, 2.77, 3.36, 3.95, 4.54, 5.13, 5.72, 6.31, 6.9 ]),
 <BarContainer object of 10 artists>)
```



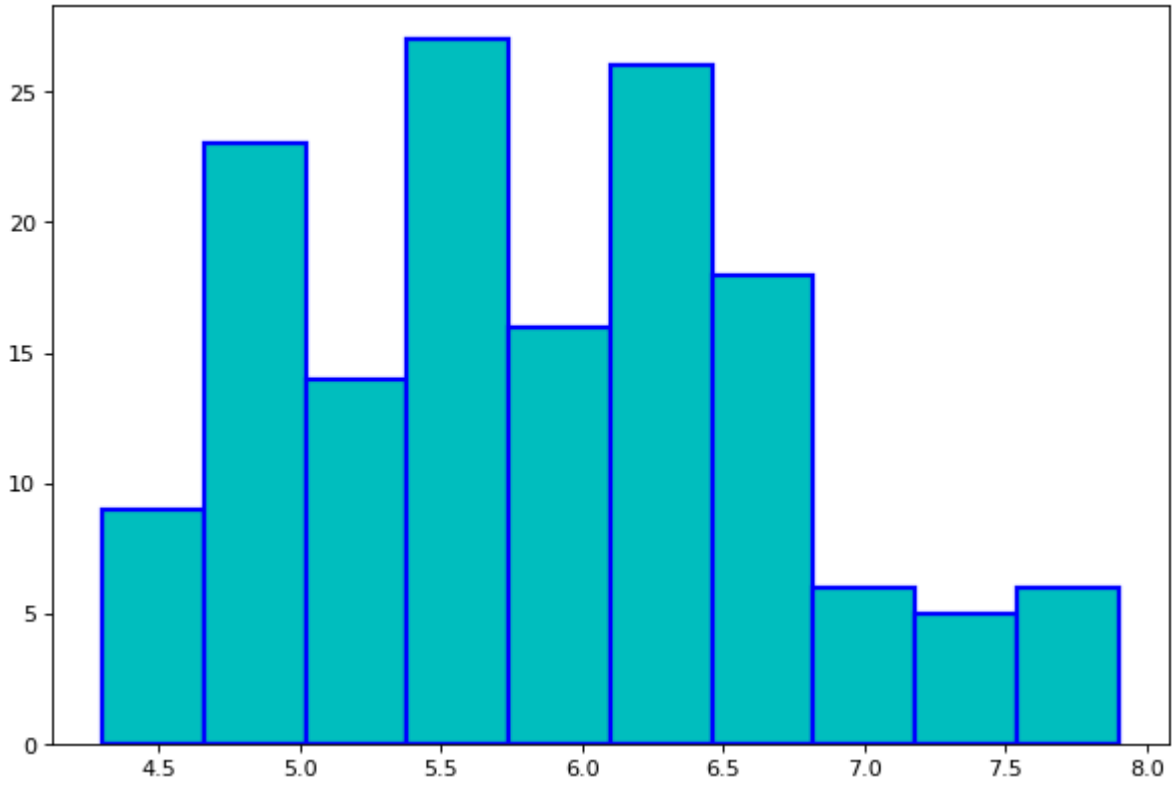
In [10]:
```
plt.hist(df['PetalWidthCm'],color='r')
```

Out[10]:
```
(array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
 array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
 <BarContainer object of 10 artists>)
```



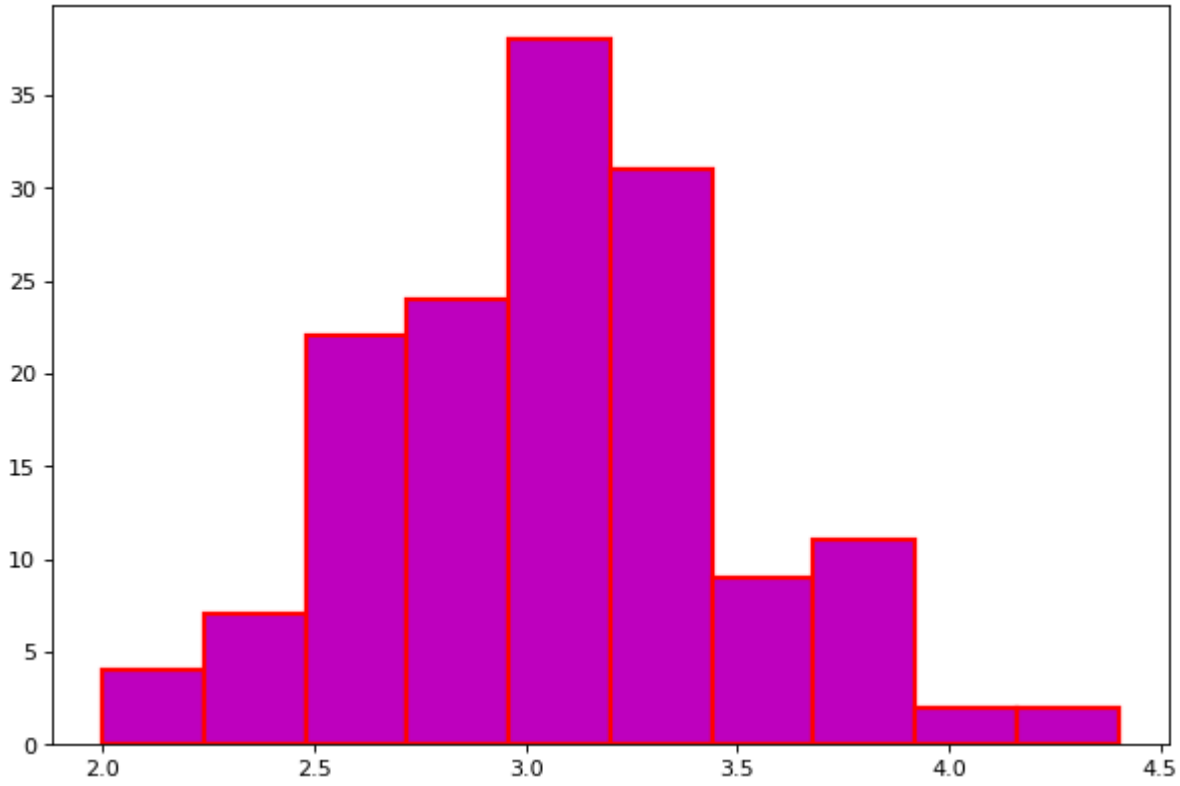
In [11]:
```
plt.hist(df['SepalLengthCm'],color='c', bins=10, linewidth=2, edgecolor="b")
```

Out[11]:
```
(array([ 9., 23., 14., 27., 16., 26., 18.,  6.,  5.,  6.]),
 array([4.3 , 4.66, 5.02, 5.38, 5.74, 6.1 , 6.46, 6.82, 7.18, 7.54, 7.9 ]),
 <BarContainer object of 10 artists>)
```



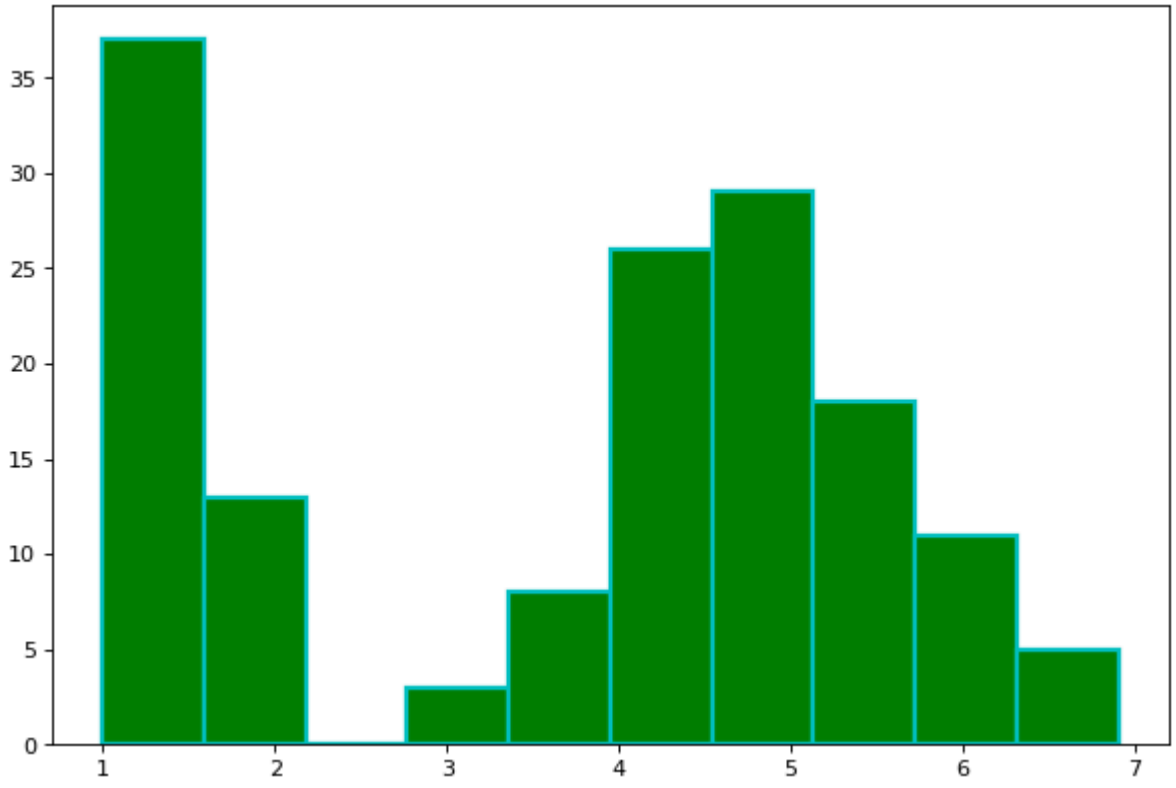
In [12]:
```
plt.hist(df['SepalWidthCm'],color='m', bins=10, linewidth=2, edgecolor="r")
```

Out[12]:
```
(array([ 4.,  7., 22., 24., 38., 31.,  9., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```



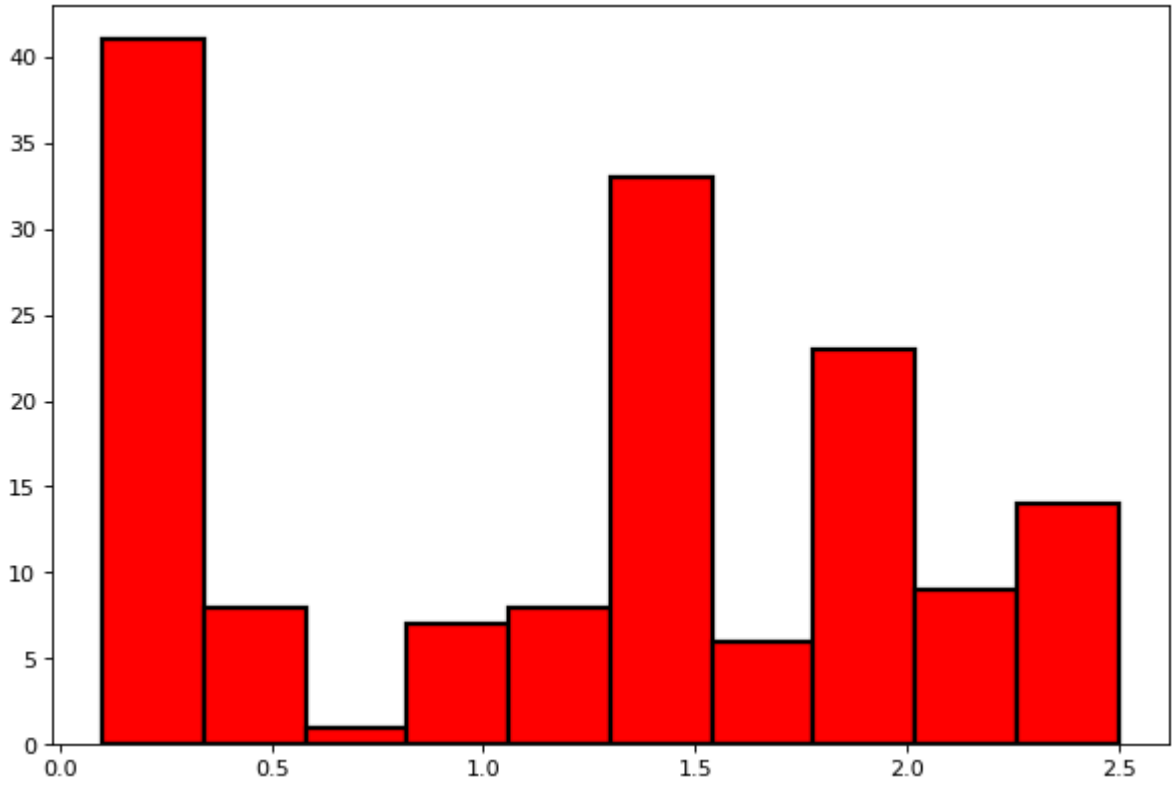
In [13]:
```
plt.hist(df['PetalLengthCm'],color='g', bins=10, linewidth=2, edgecolor="c")
```

Out[13]:
```
(array([37., 13.,  0.,  3.,  8., 26., 29., 18., 11.,  5.]),
 array([1.  , 1.59, 2.18, 2.77, 3.36, 3.95, 4.54, 5.13, 5.72, 6.31, 6.9 ]),
 <BarContainer object of 10 artists>)
```



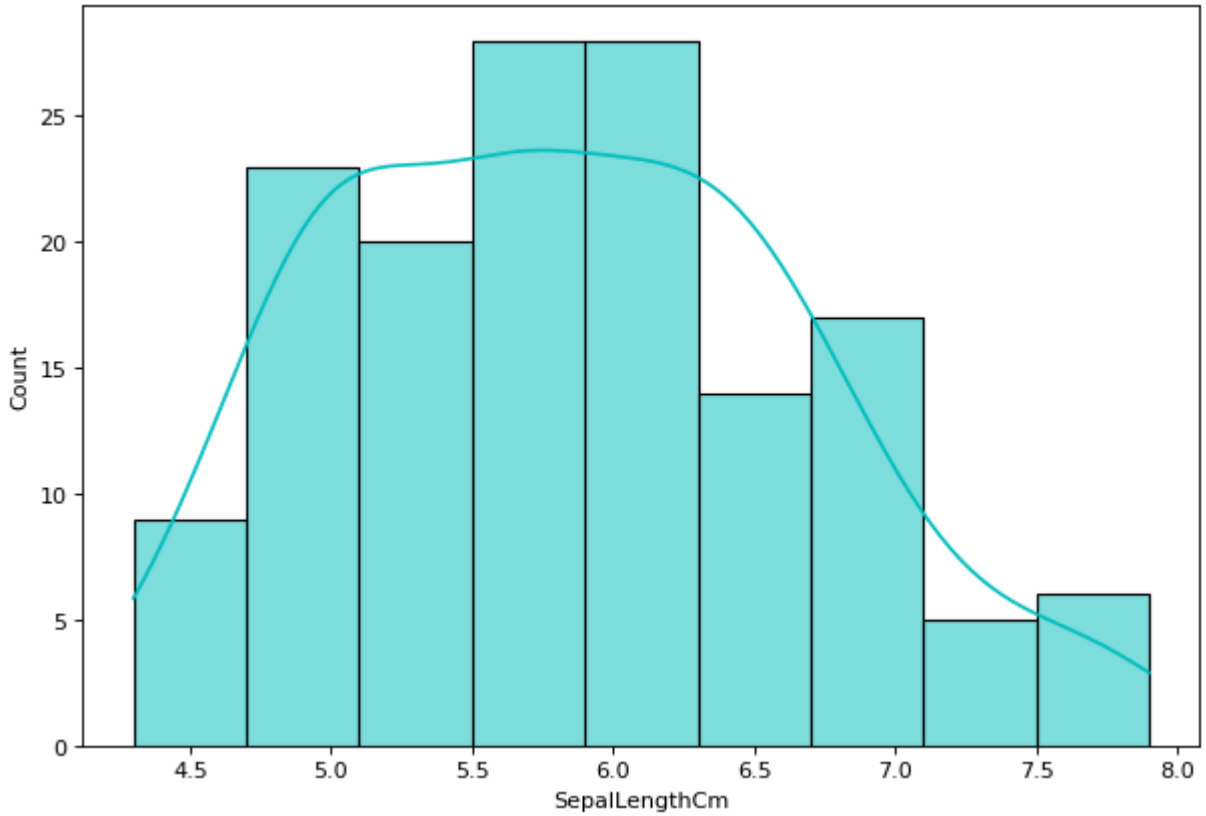
In [14]:
```
plt.hist(df['PetalWidthCm'],color='r', bins=10, linewidth=2, edgecolor="black")
```

Out[14]:
```
(array([41.,  8.,  1.,  7.,  8., 33.,  6., 23.,  9., 14.]),
 array([0.1 , 0.34, 0.58, 0.82, 1.06, 1.3 , 1.54, 1.78, 2.02, 2.26, 2.5 ]),
 <BarContainer object of 10 artists>)
```



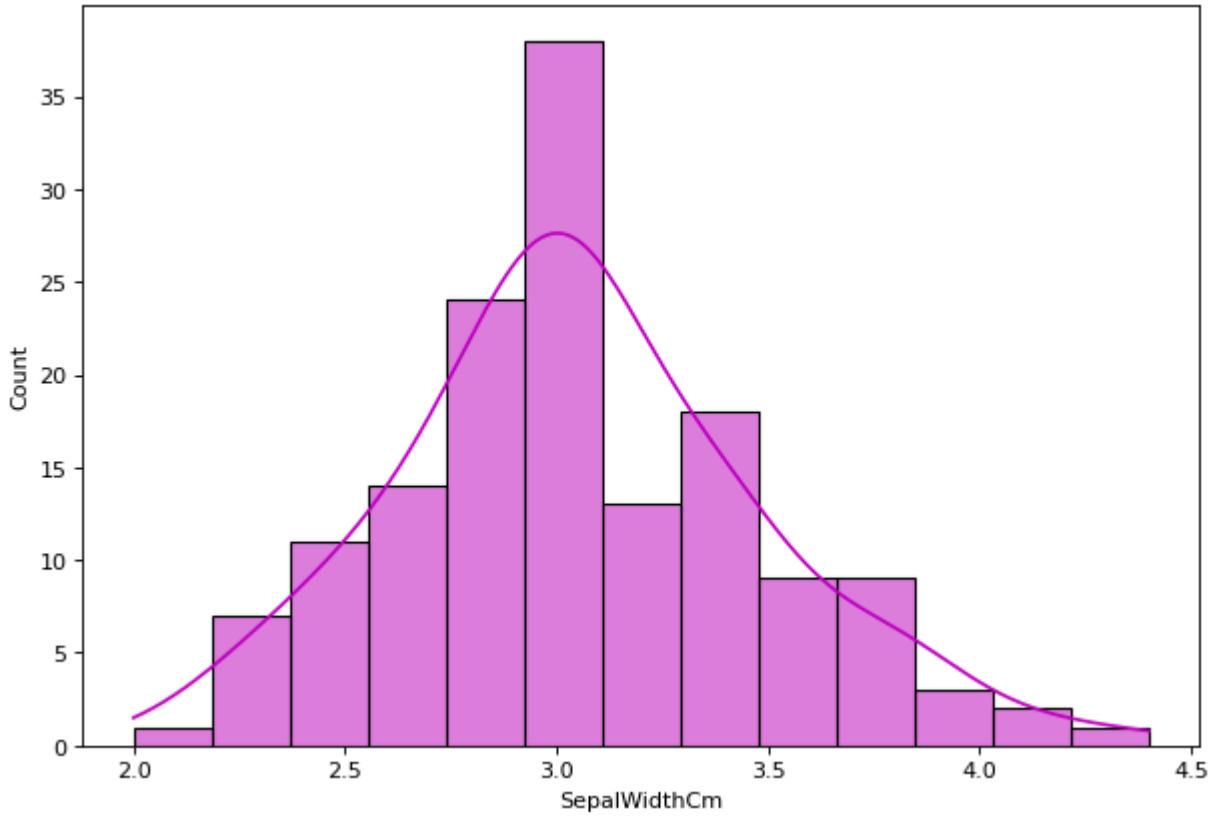
In [15]:
```
sns.histplot(data=df['SepalLengthCm'],color='c',kde=True)
```

Out[15]:
```
<Axes: xlabel='SepalLengthCm', ylabel='Count'>
```



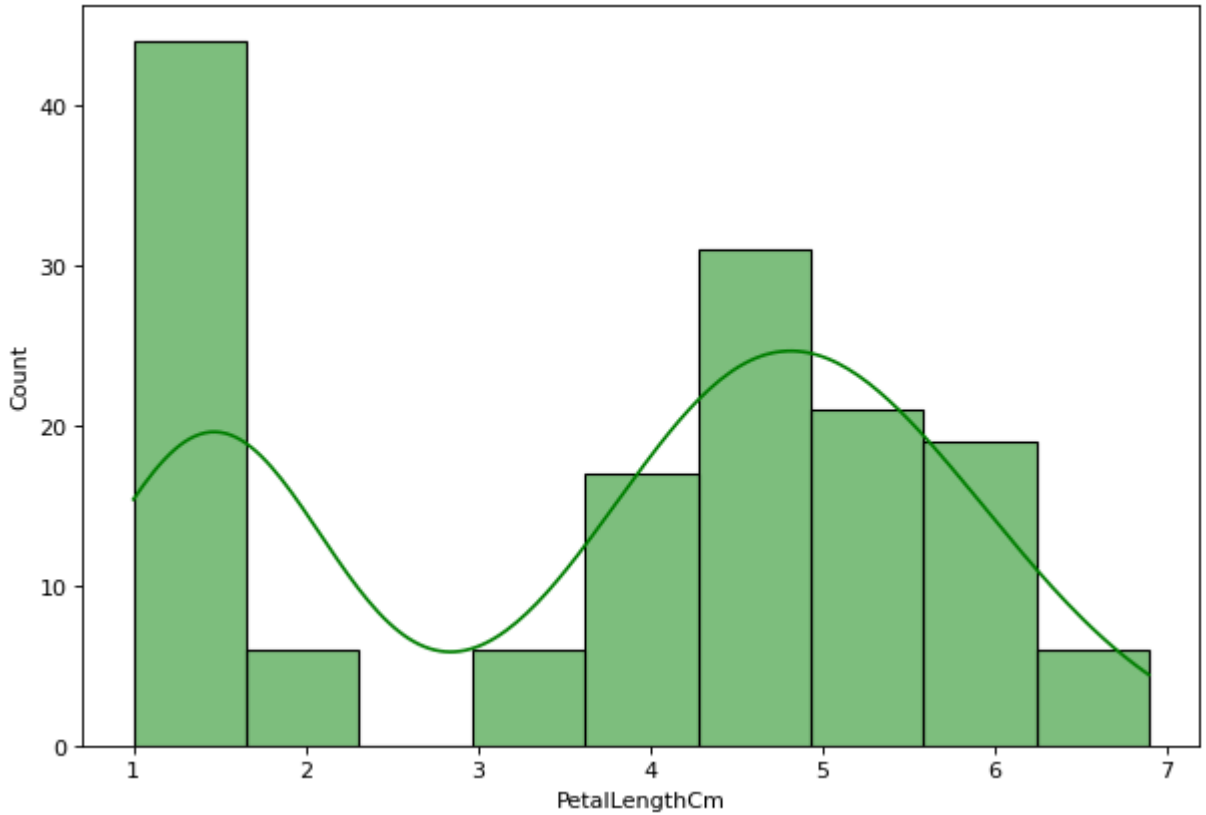
In [16]:
```
sns.histplot(data=df['SepalWidthCm'],color='m',kde=True)
```

Out[16]:
```
<Axes: xlabel='SepalWidthCm', ylabel='Count'>
```



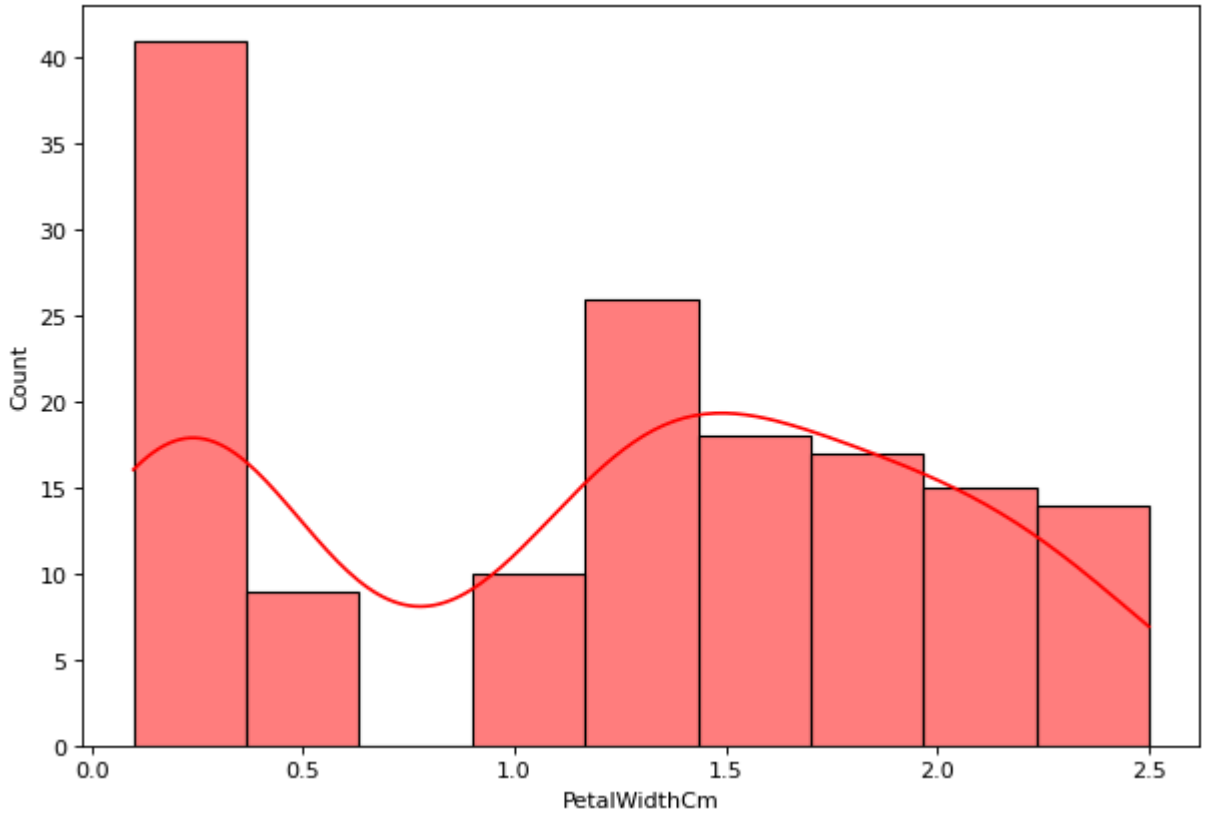
In [17]:
```
sns.histplot(data=df['PetalLengthCm'],color='g',kde=True)
```

Out[17]:
```
<Axes: xlabel='PetalLengthCm', ylabel='Count'>
```

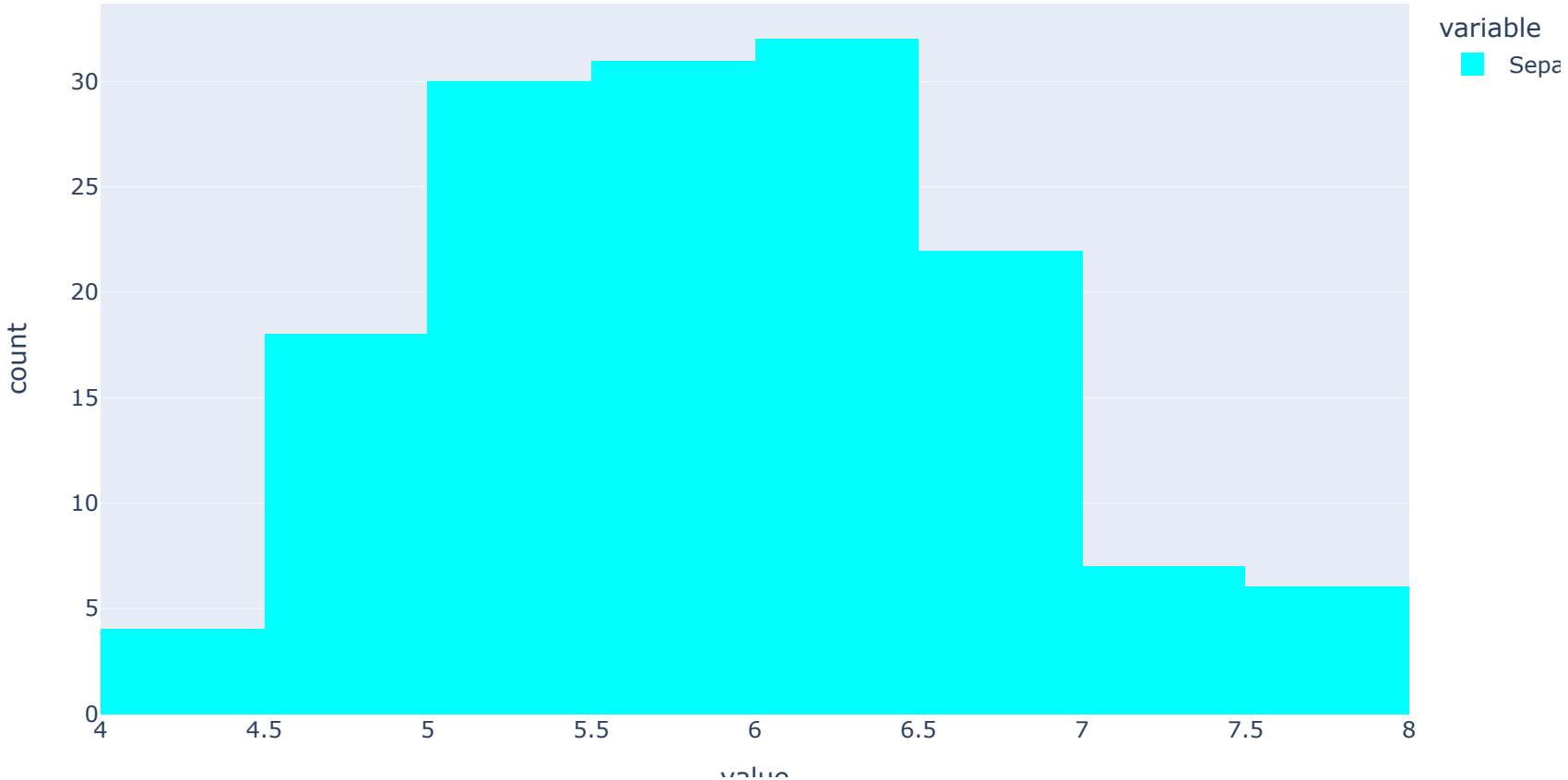


In [18]:
```
sns.histplot(data=df['PetalWidthCm'],color='r',kde=True)
```

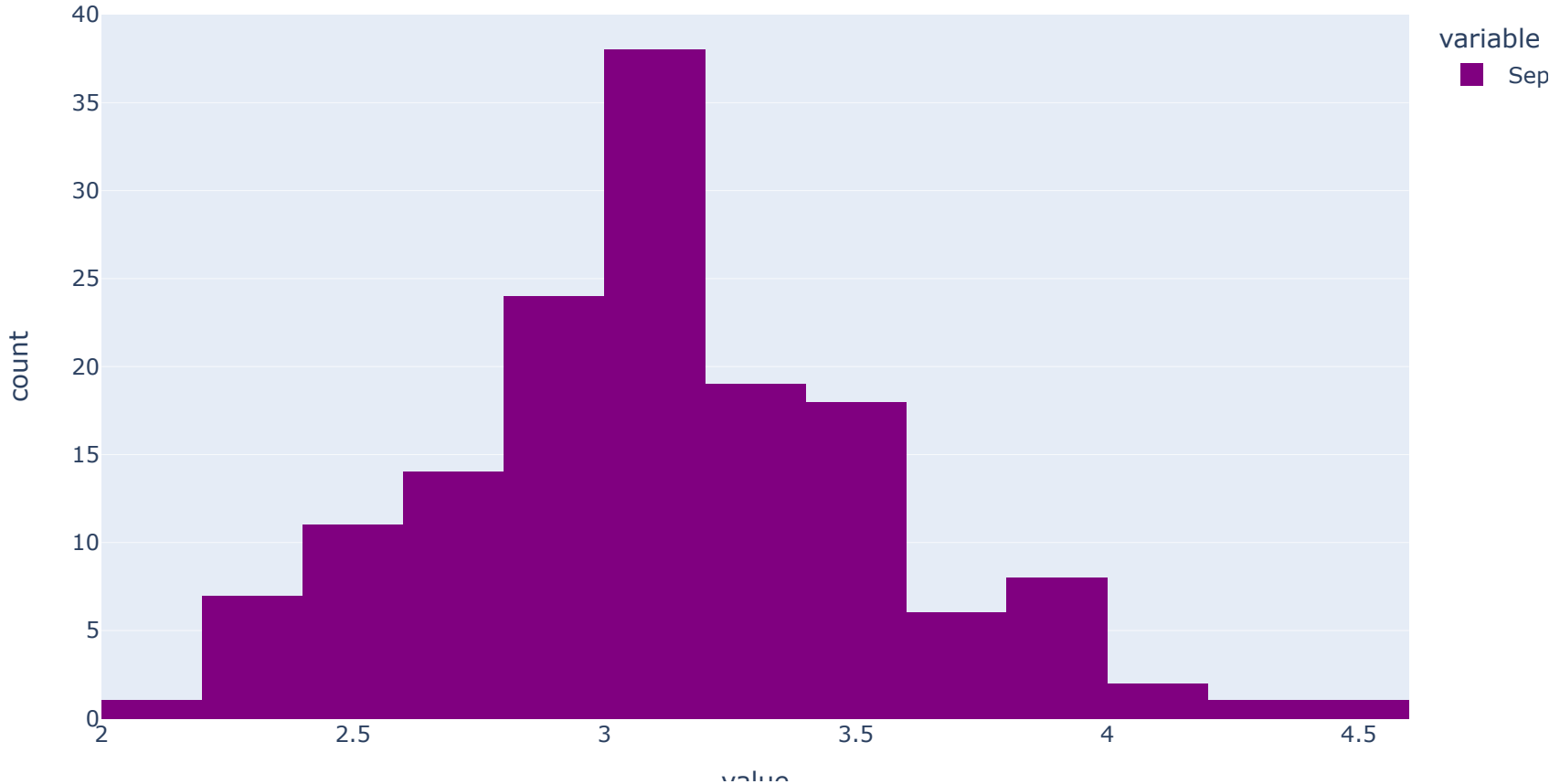
Out[18]:
```
<Axes: xlabel='PetalWidthCm', ylabel='Count'>
```



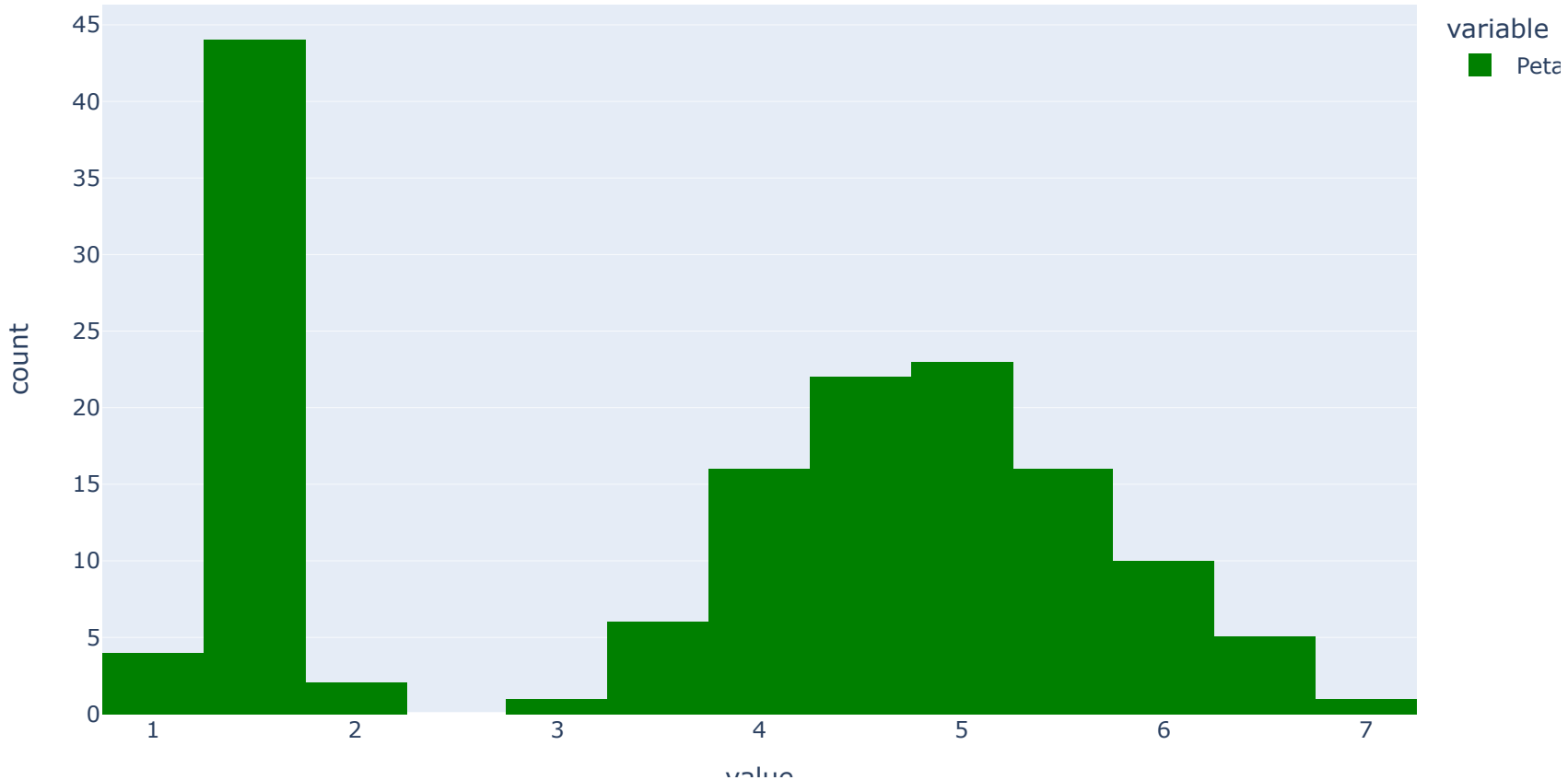
In [19]:
```
import plotly.express as px

px.histogram(df['SepalLengthCm'],color_discrete_sequence=['cyan'])
```



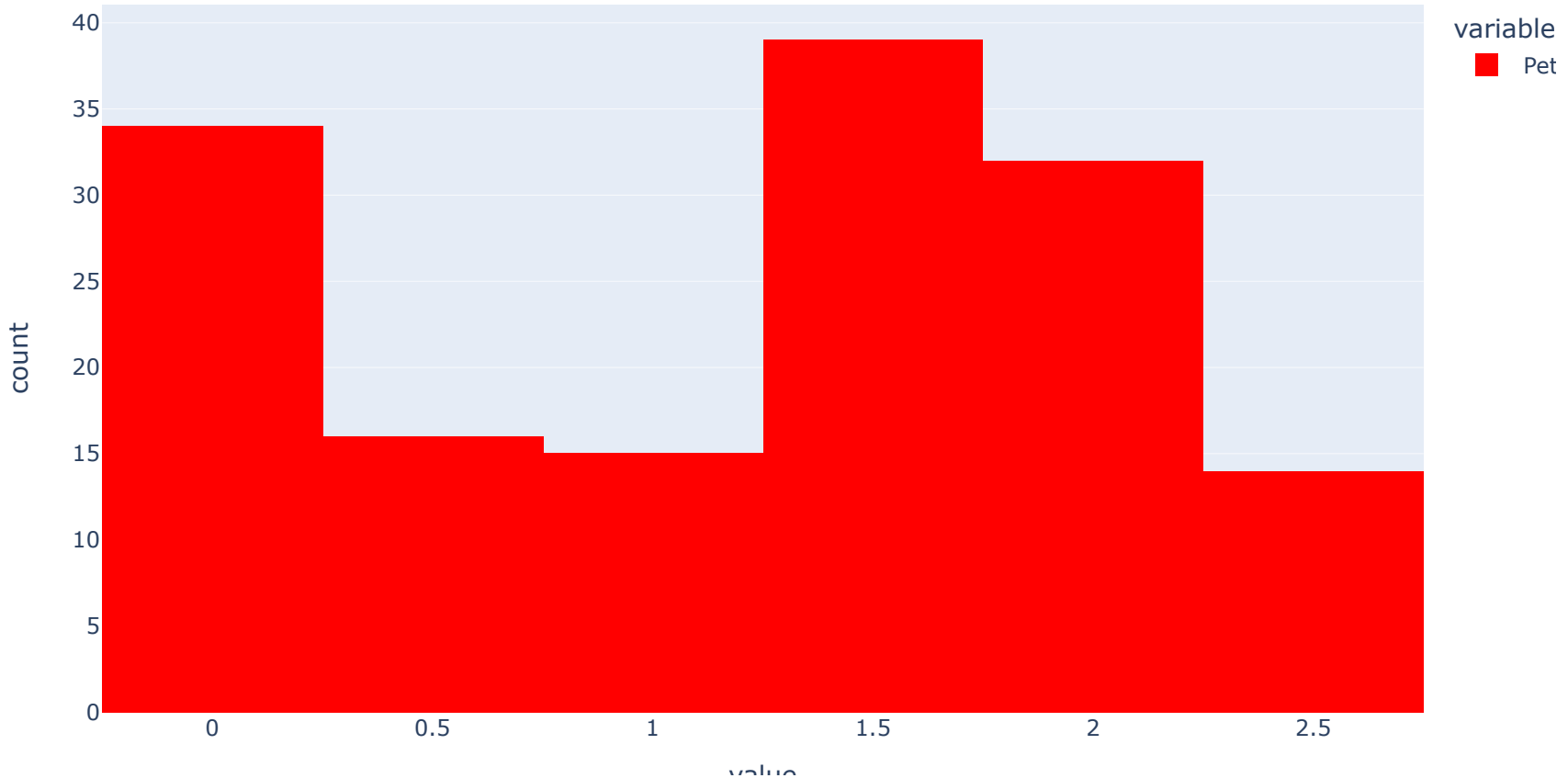
In [20]:
```
px.histogram(df['SepalWidthCm'],color_discrete_sequence=['purple'])
```



In [21]:
```
px.histogram(df['PetalLengthCm'],color_discrete_sequence=['green'])
```

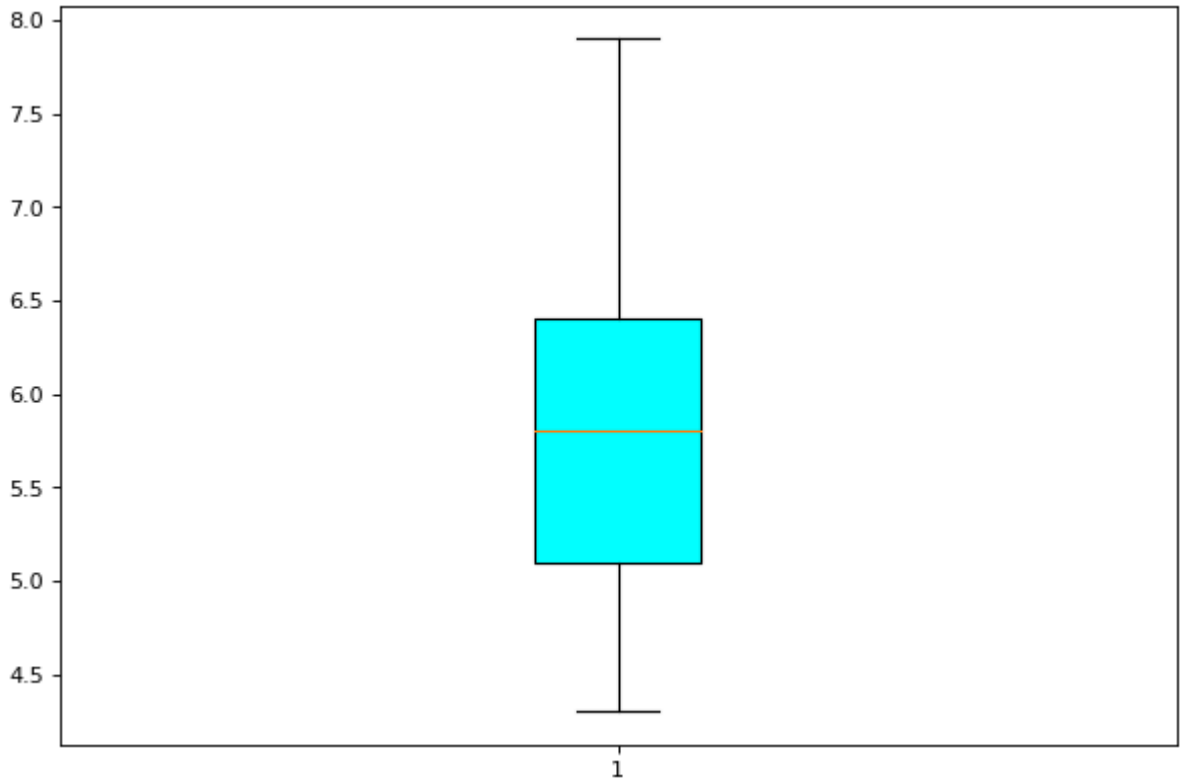


In [22]:
```
px.histogram(df['PetalWidthCm'],color_discrete_sequence=['red'])
```

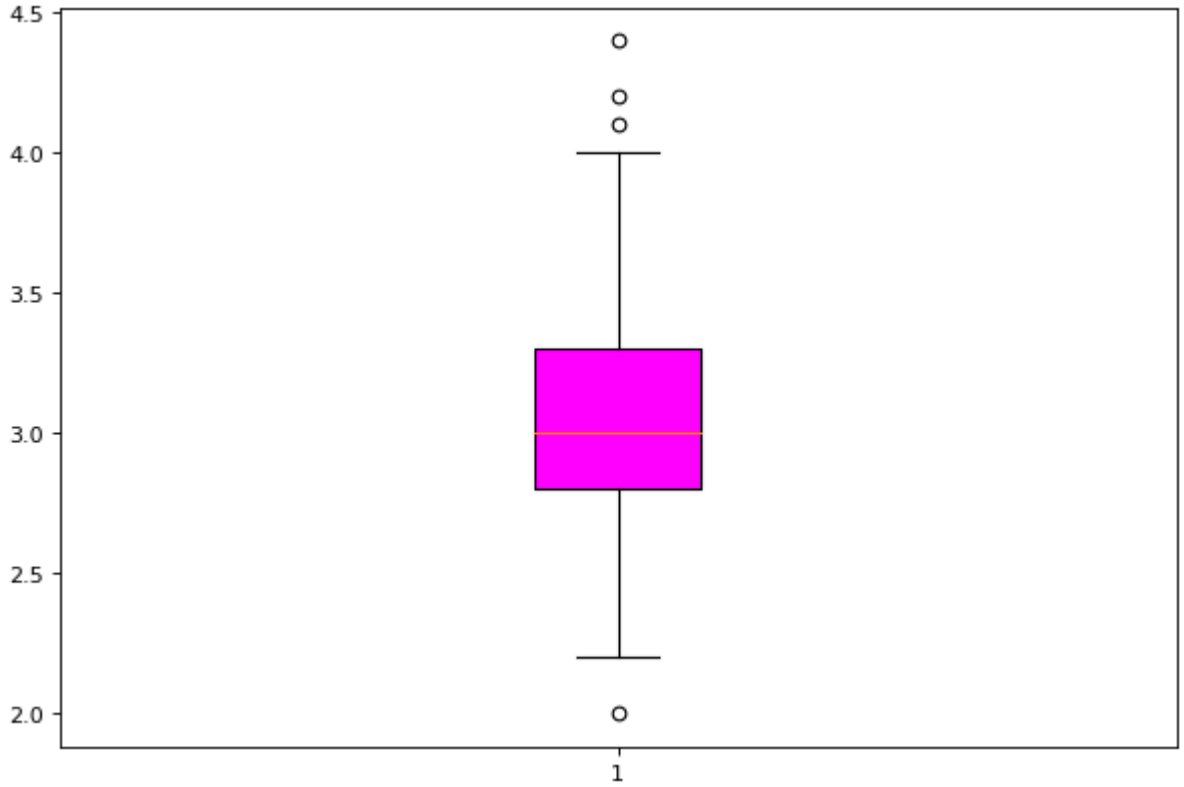


In [23]:
```
#Plotting Box Plot for SepalLength, SepalWidth, PetalLength, PetalWidth
```

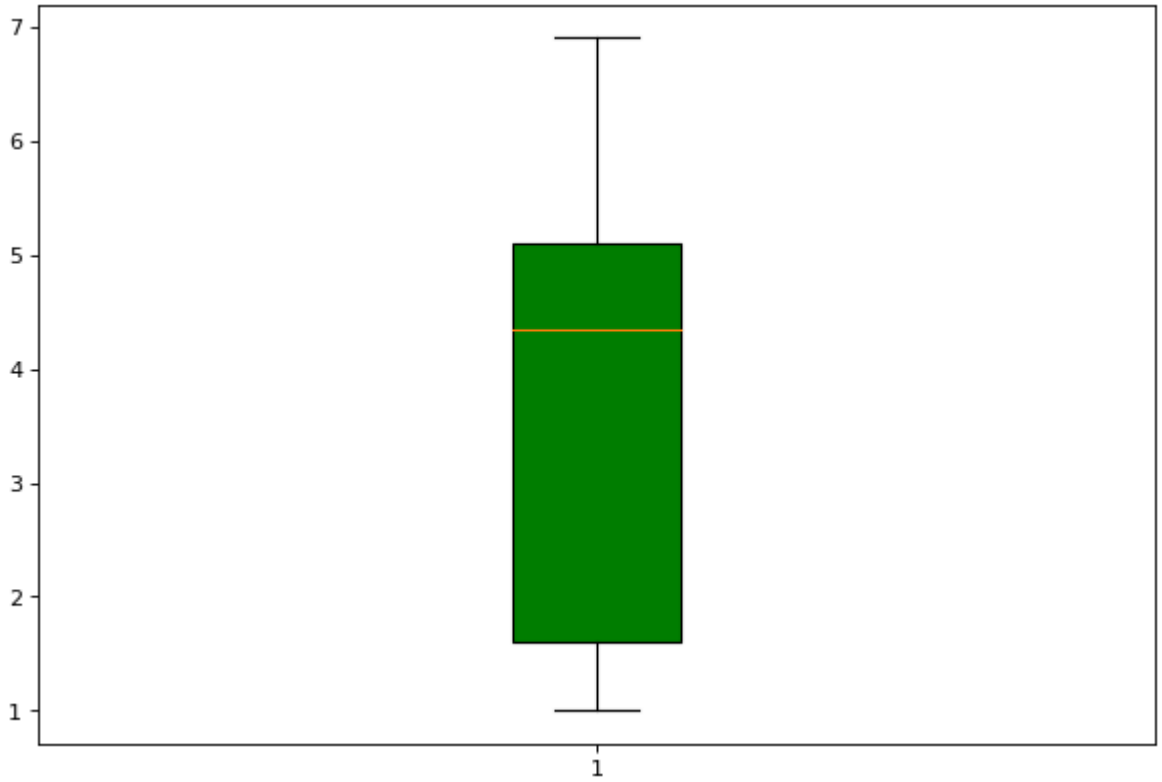
In [24]:
```
plt.boxplot(df['SepalLengthCm'],patch_artist = True, boxprops = dict(facecolor = "cyan"))
plt.show()
```



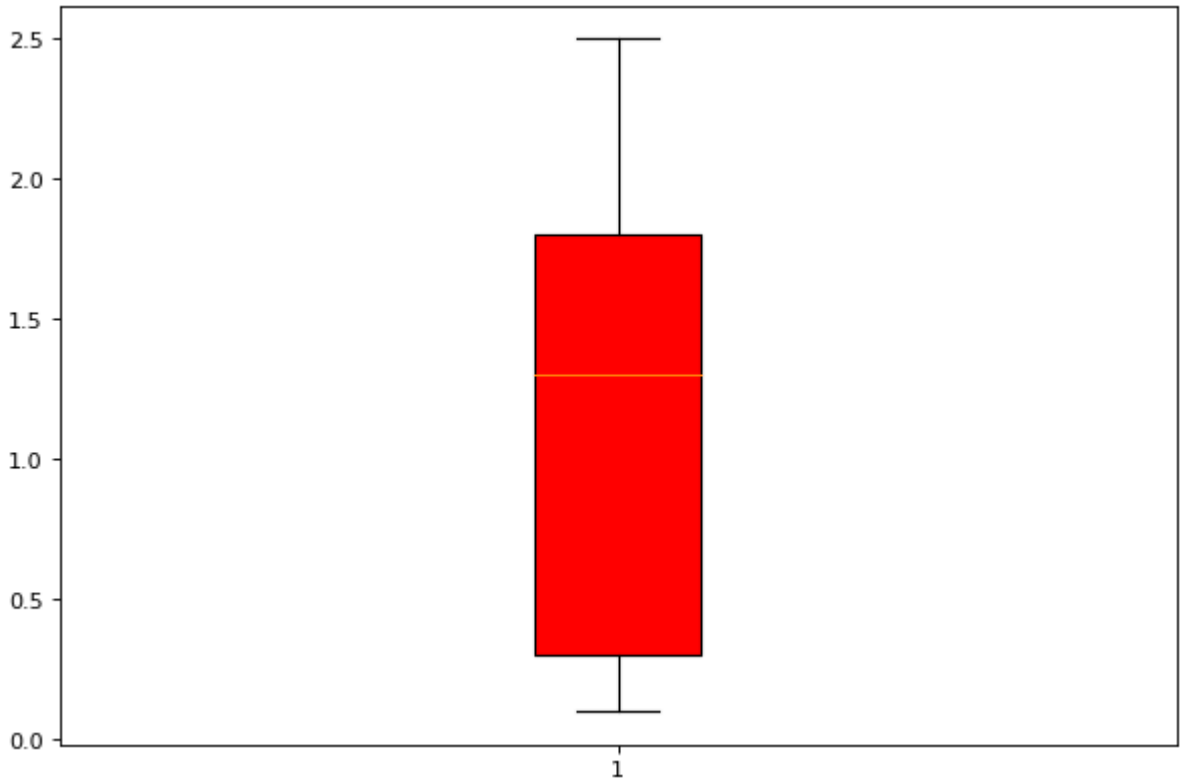
In [25]:
```
plt.boxplot(df['SepalWidthCm'],patch_artist = True, boxprops = dict(facecolor = "magenta"))
plt.show()
```



In [26]:
```
plt.boxplot(df['PetalLengthCm'],patch_artist = True, boxprops = dict(facecolor = "green"))
plt.show()
```



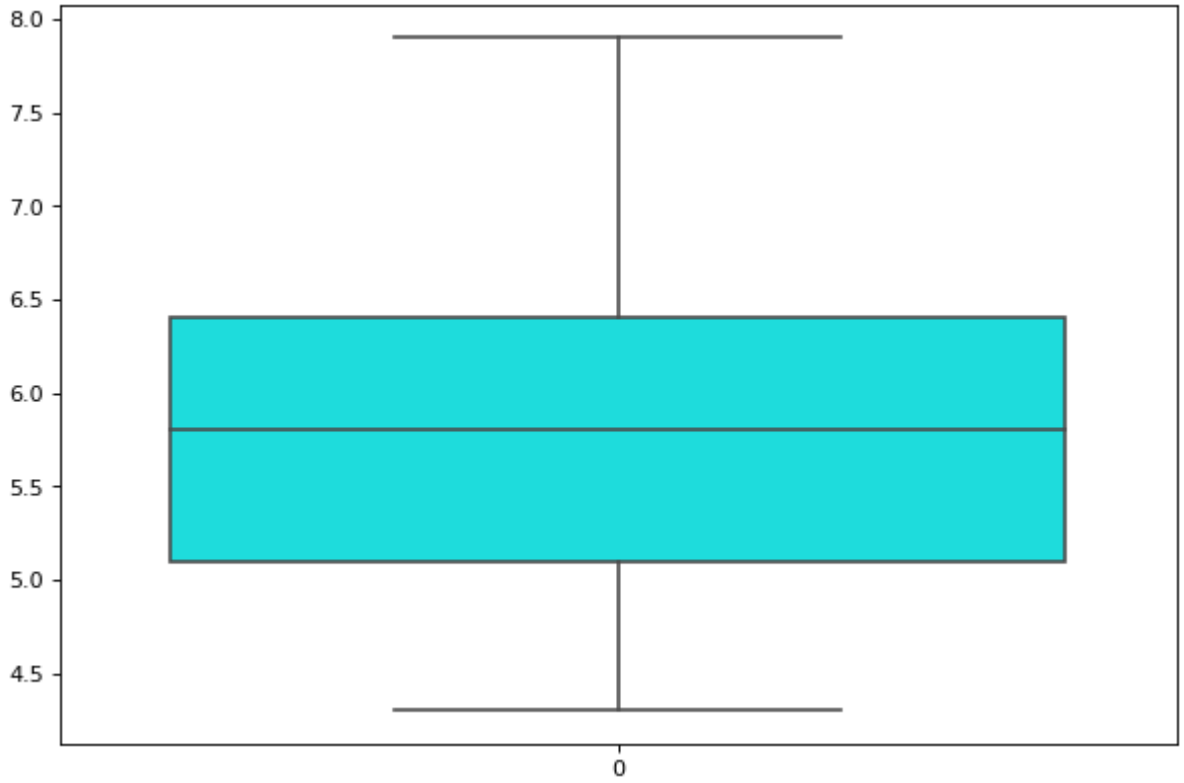
In [27]:
```
plt.boxplot(df['PetalWidthCm'],patch_artist = True, boxprops = dict(facecolor = "red"))
plt.show()
```



In [28]:
```
#Plotting Box Plot for SepalLength, SepalWidth, PetalLength, PetalWidth using seaborne library
```

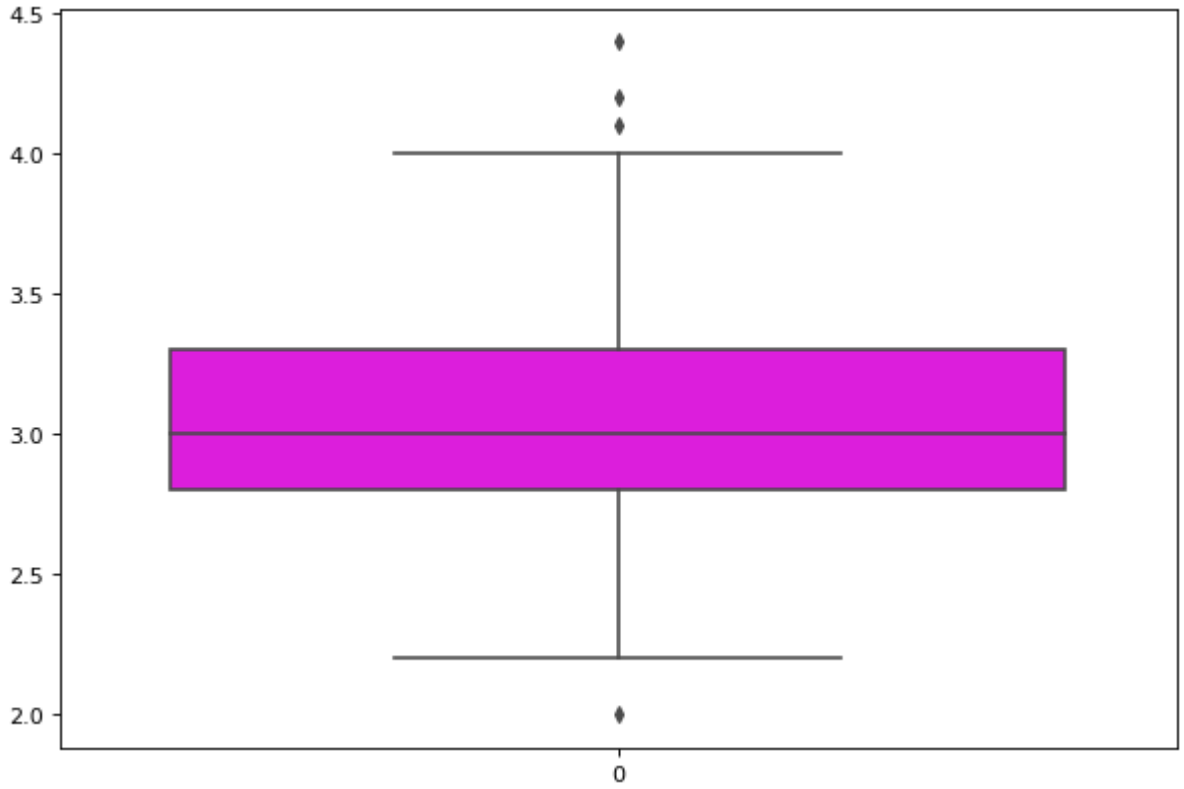
In [29]:
```
sns.boxplot(df['SepalLengthCm'],color='cyan')
```

Out[29]: <Axes: >



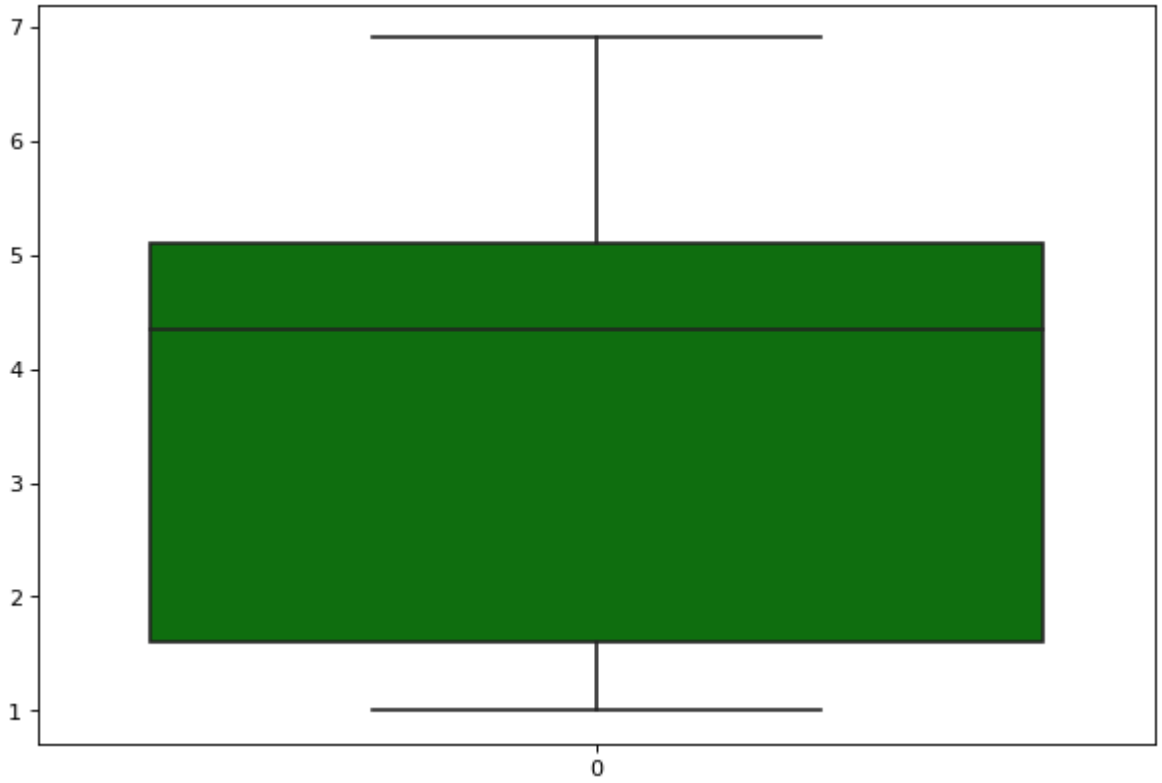
In [30]:
```
sns.boxplot(df['SepalWidthCm'],color='magenta')
```

Out[30]: <Axes: >



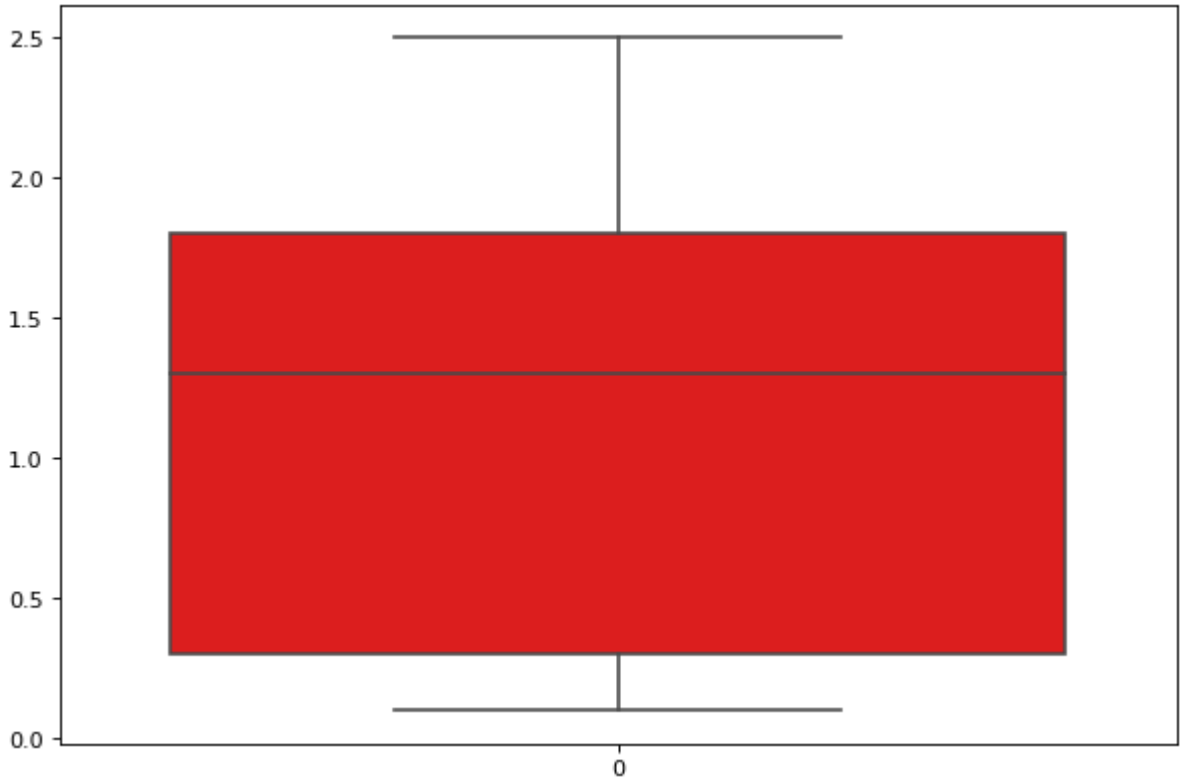
In [31]:
```
sns.boxplot(df['PetalLengthCm'],color='green')
```

Out[31]: <Axes: >



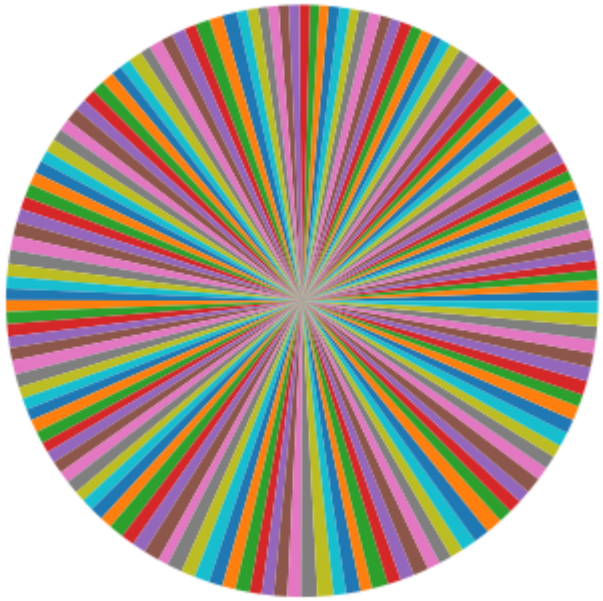
In [32]:
```
sns.boxplot(df['PetalWidthCm'],color='red')
```

Out[32]: <Axes: >

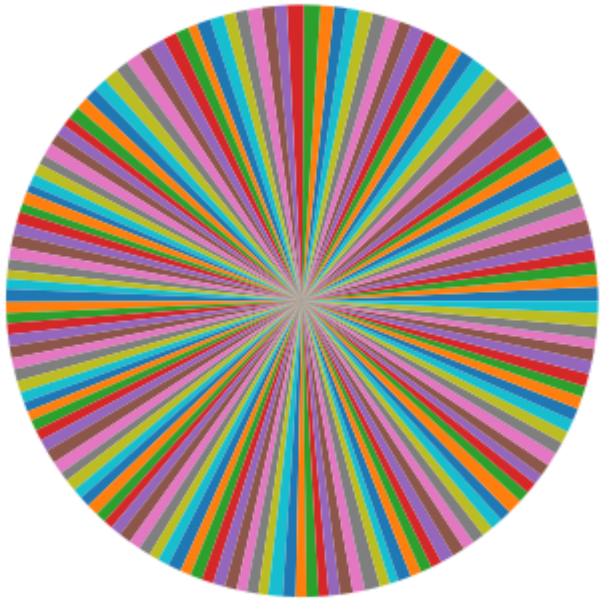


In [33]:
```
#Plotting Pie Plot for SepalLength, SepalWidth, PetalLength, PetalWidth
```

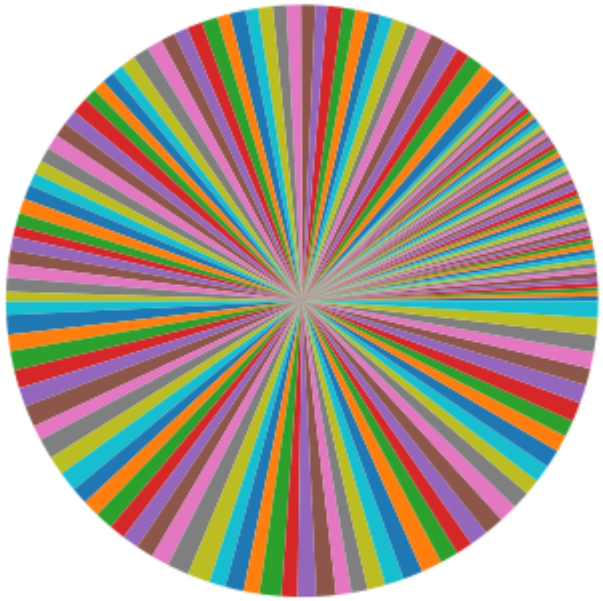
In [34]:
```
plt.pie(df['SepalLengthCm'])
plt.show()
```



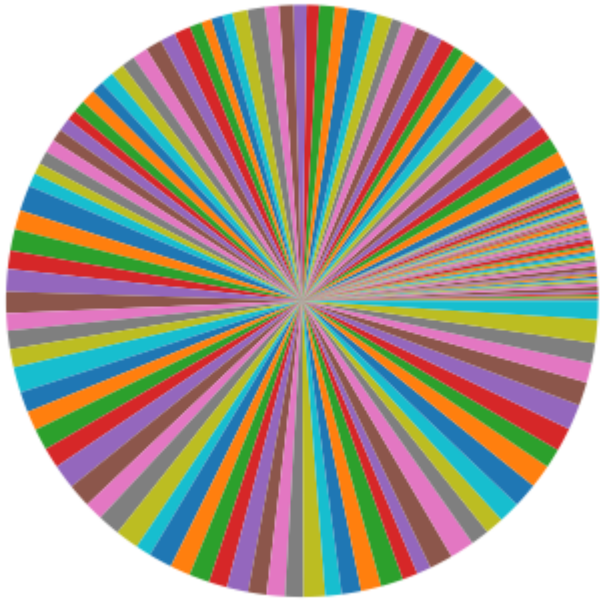
In [35]:
```
plt.pie(df['SepalWidthCm'])
plt.show()
```



In [36]:
```
plt.pie(df['PetalLengthCm'])
plt.show()
```



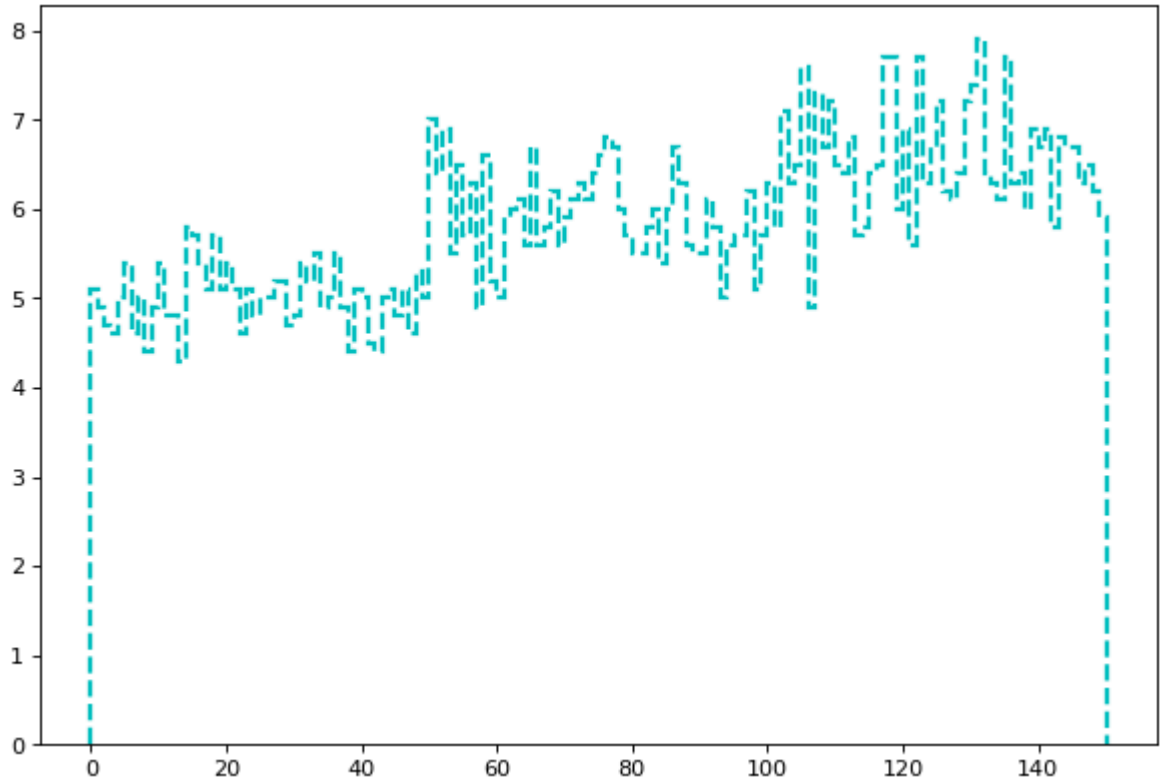
In [37]:
```
plt.pie(df['PetalWidthCm'])
plt.show()
```



In [38]:
```
#Plotting Stair Plot for SepalLength, SepalWidth, PetalLength, PetalWidth
```

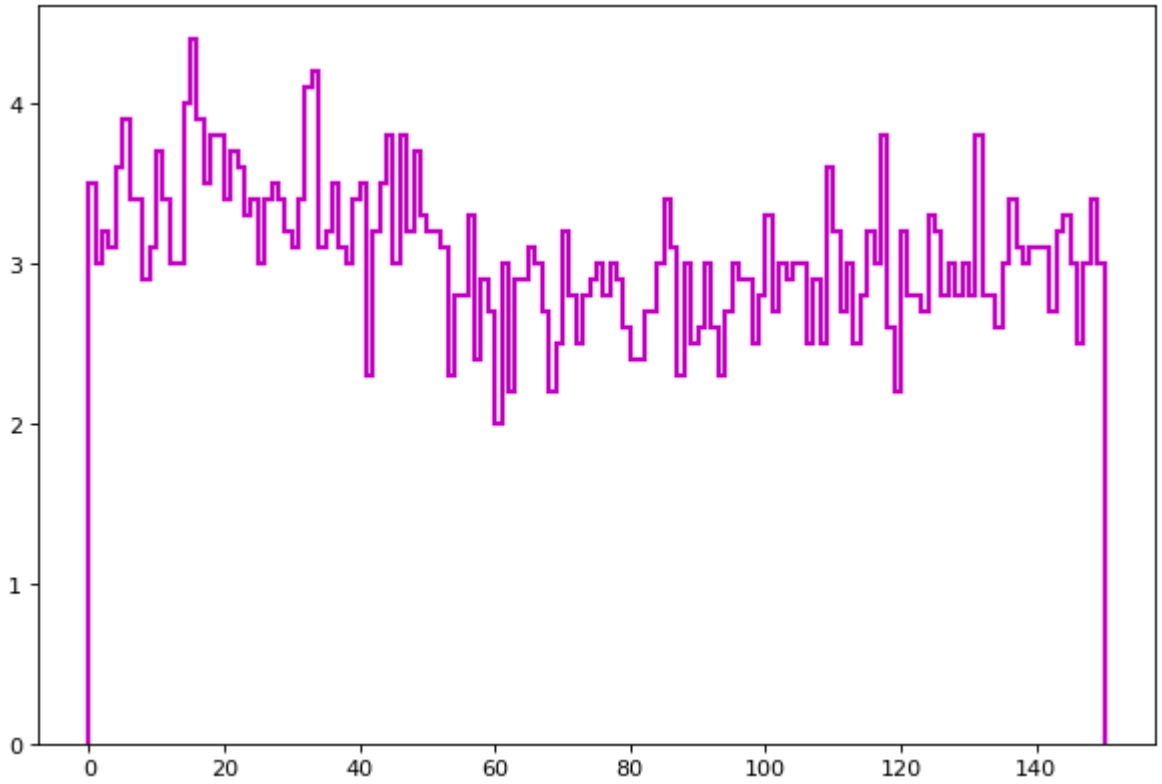
In [39]:
```
plt.stairs(df['SepalLengthCm'], linewidth=2, linestyle = '--', color='c')
```

Out[39]:
```
<matplotlib.patches.StepPatch at 0x25fc1486790>
```



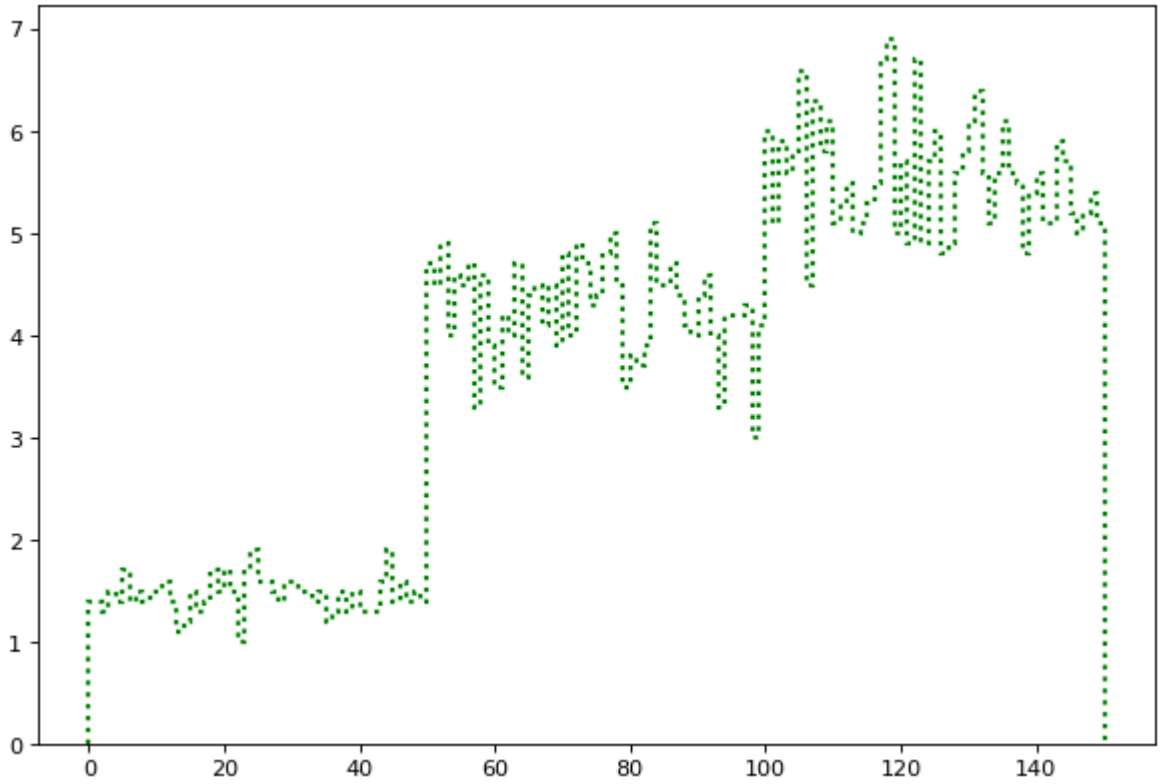
In [40]:
```
plt.stairs(df['SepalWidthCm'], linewidth=2, linestyle = '-', color='m')
```

Out[40]:
```
<matplotlib.patches.StepPatch at 0x25fc10ee790>
```



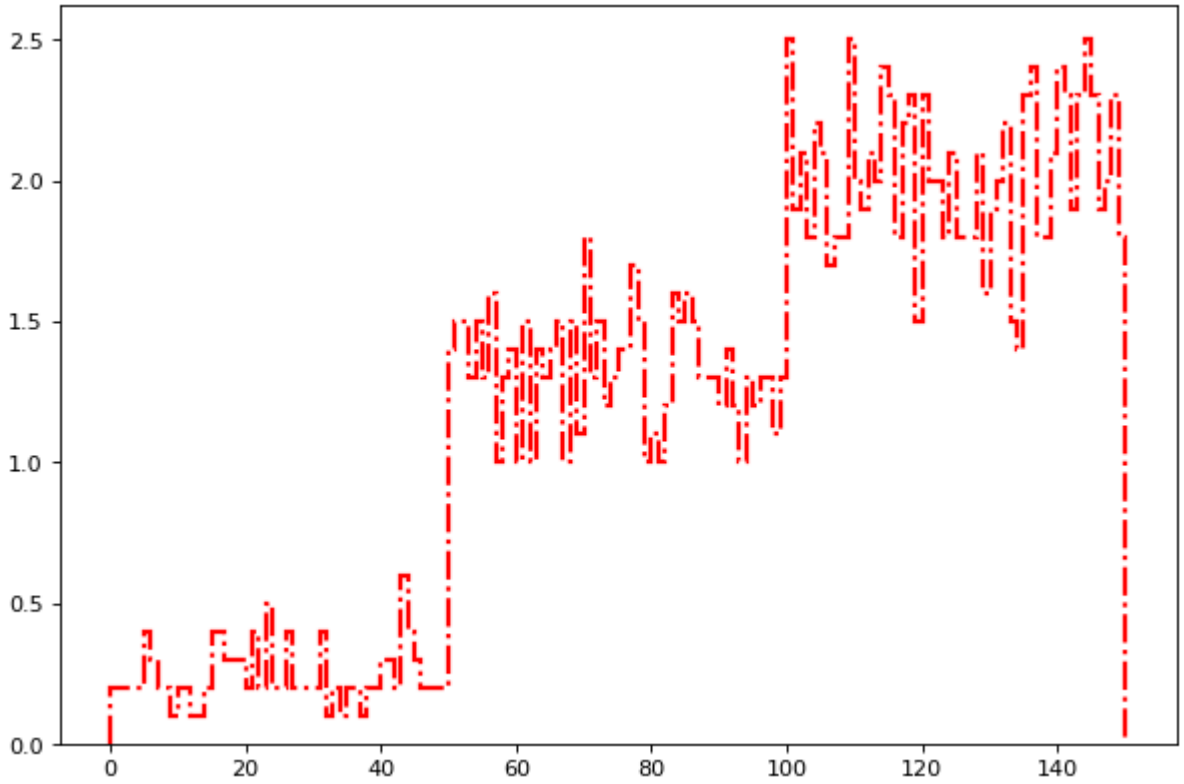
In [41]:
```
plt.stairs(df['PetalLengthCm'], linewidth=2, linestyle = ':', color='g')
```

Out[41]:
```
<matplotlib.patches.StepPatch at 0x25fc11c3c50>
```



In [42]:
```
plt.stairs(df['PetalWidthCm'], linewidth=2, linestyle = '-.' , color='r')
```

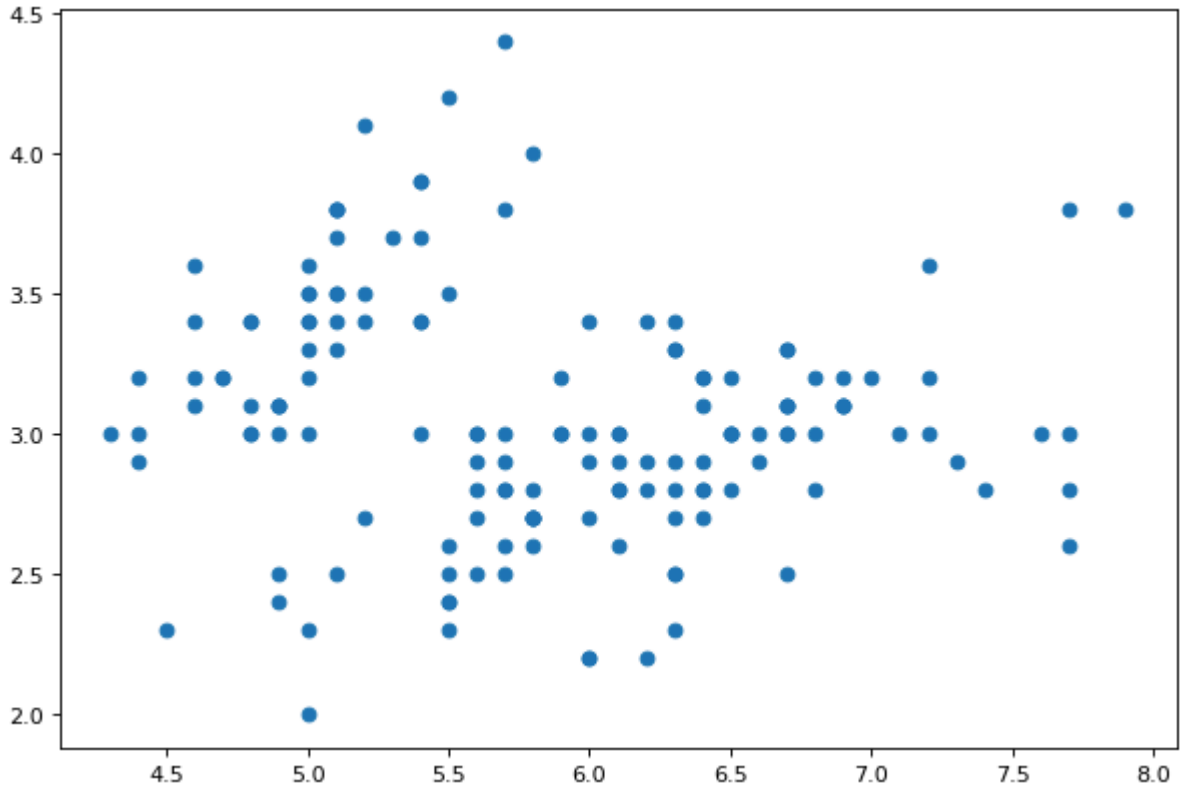
Out[42]:
```
<matplotlib.patches.StepPatch at 0x25fbff16390>
```



In [43]:
```
#Plotting Scatter Plot for SepalLength, SepalWidth, PetalLength, PetalWidth
```

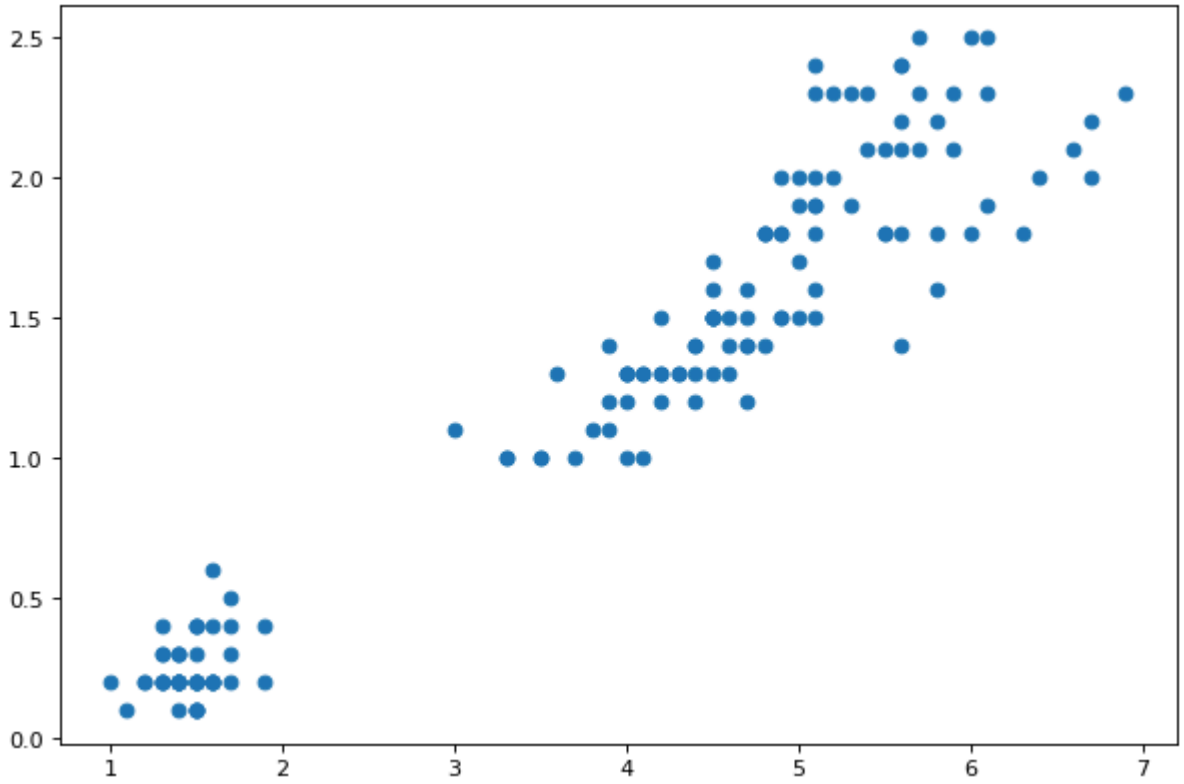
In [44]:
```
plt.scatter(df['SepalLengthCm'], df['SepalWidthCm'])
```

Out[44]:
```
<matplotlib.collections.PathCollection at 0x25fbff43f10>
```



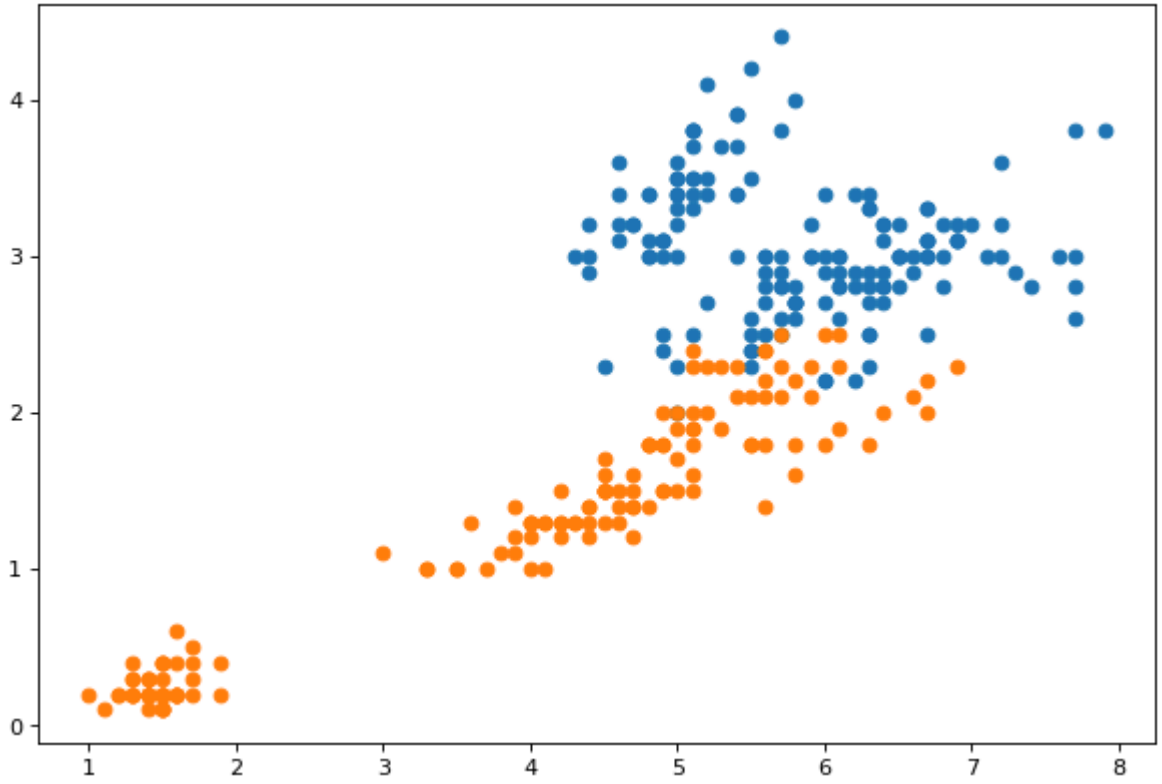
In [45]:
```
plt.scatter(df['PetalLengthCm'], df['PetalWidthCm'])
```

Out[45]:
```
<matplotlib.collections.PathCollection at 0x25fc179c5d0>
```

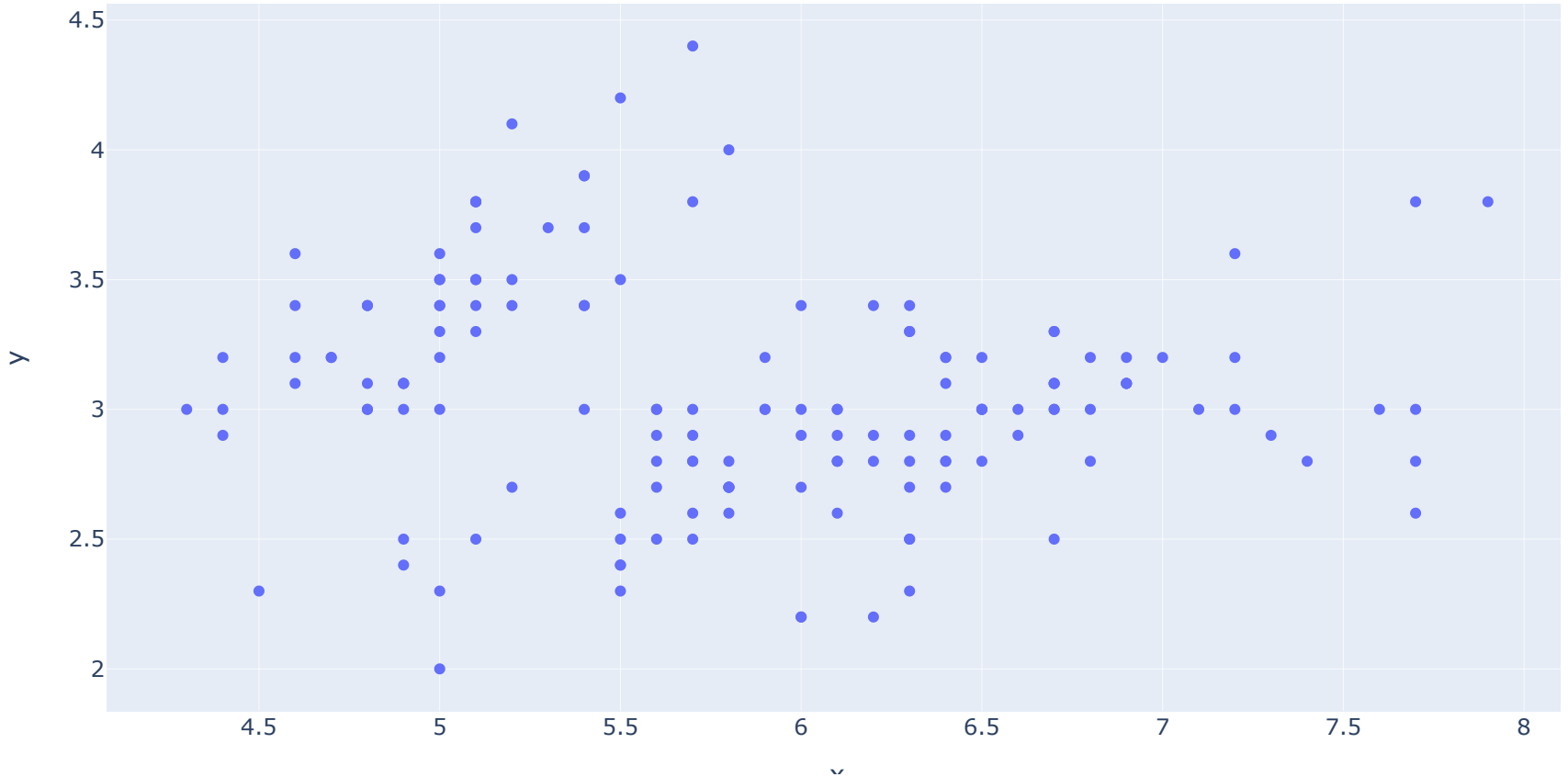


In [46]:
```
plt.scatter(df['SepalLengthCm'], df['SepalWidthCm'])
plt.scatter(df['PetalLengthCm'], df['PetalWidthCm'])
```

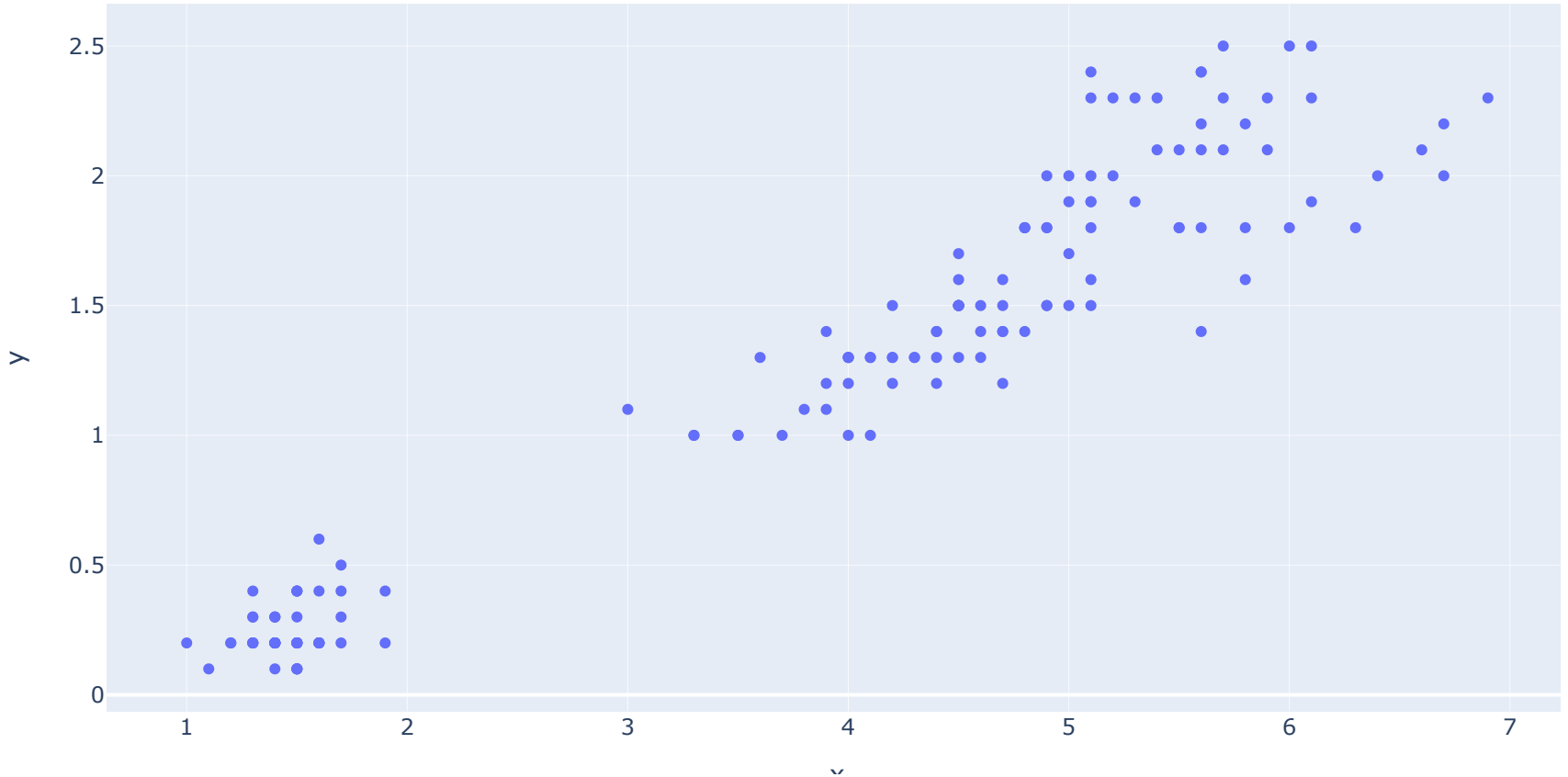
Out[46]: <matplotlib.collections.PathCollection at 0x25fc182ddd0>



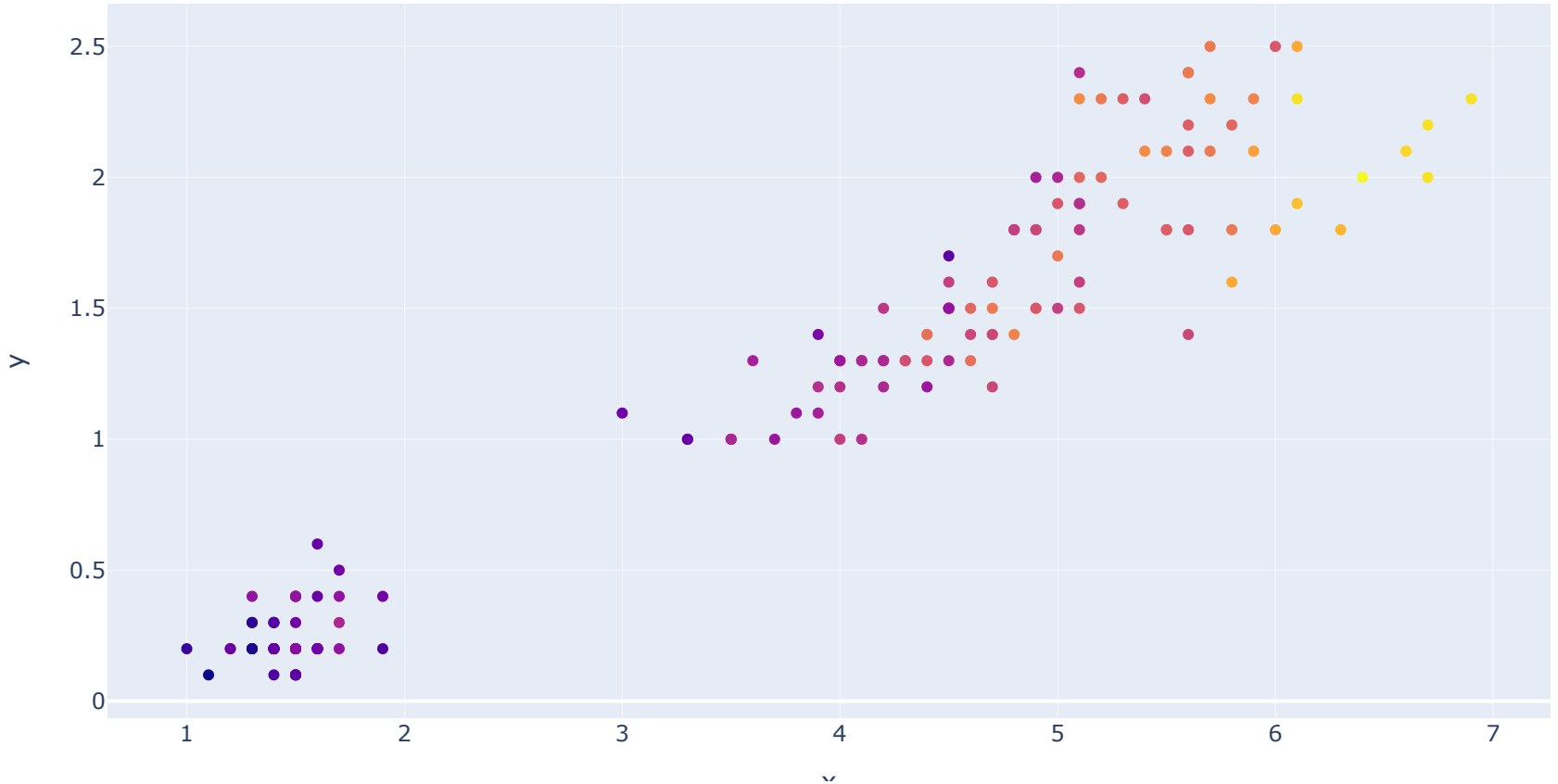
In [47]:
```
fig = px.scatter(x=df['SepalLengthCm'], y=df['SepalWidthCm'])
fig.show()
```



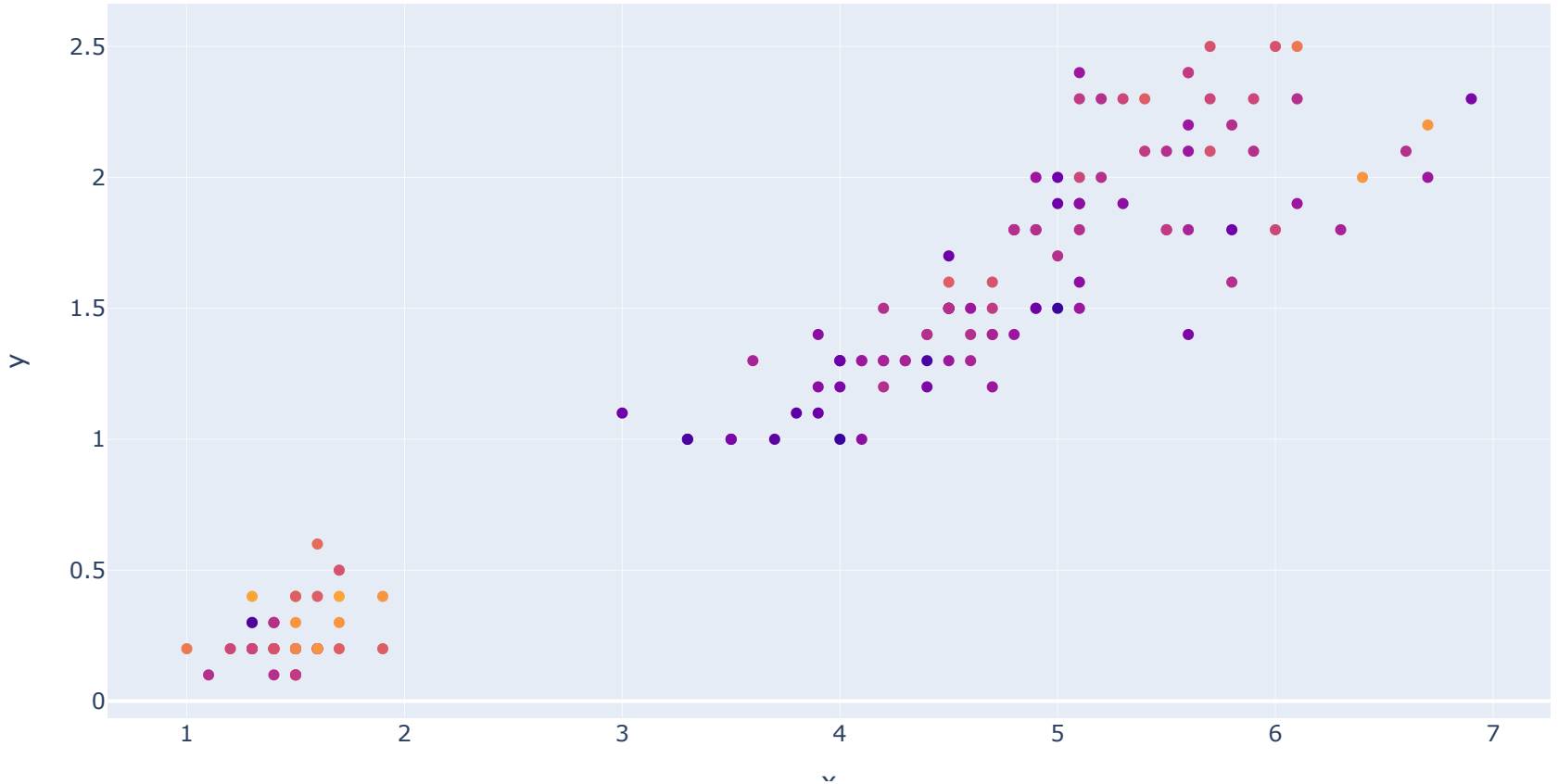
In [48]:
```
fig = px.scatter(x=df['PetalLengthCm'], y=df['PetalWidthCm'])
fig.show()
```



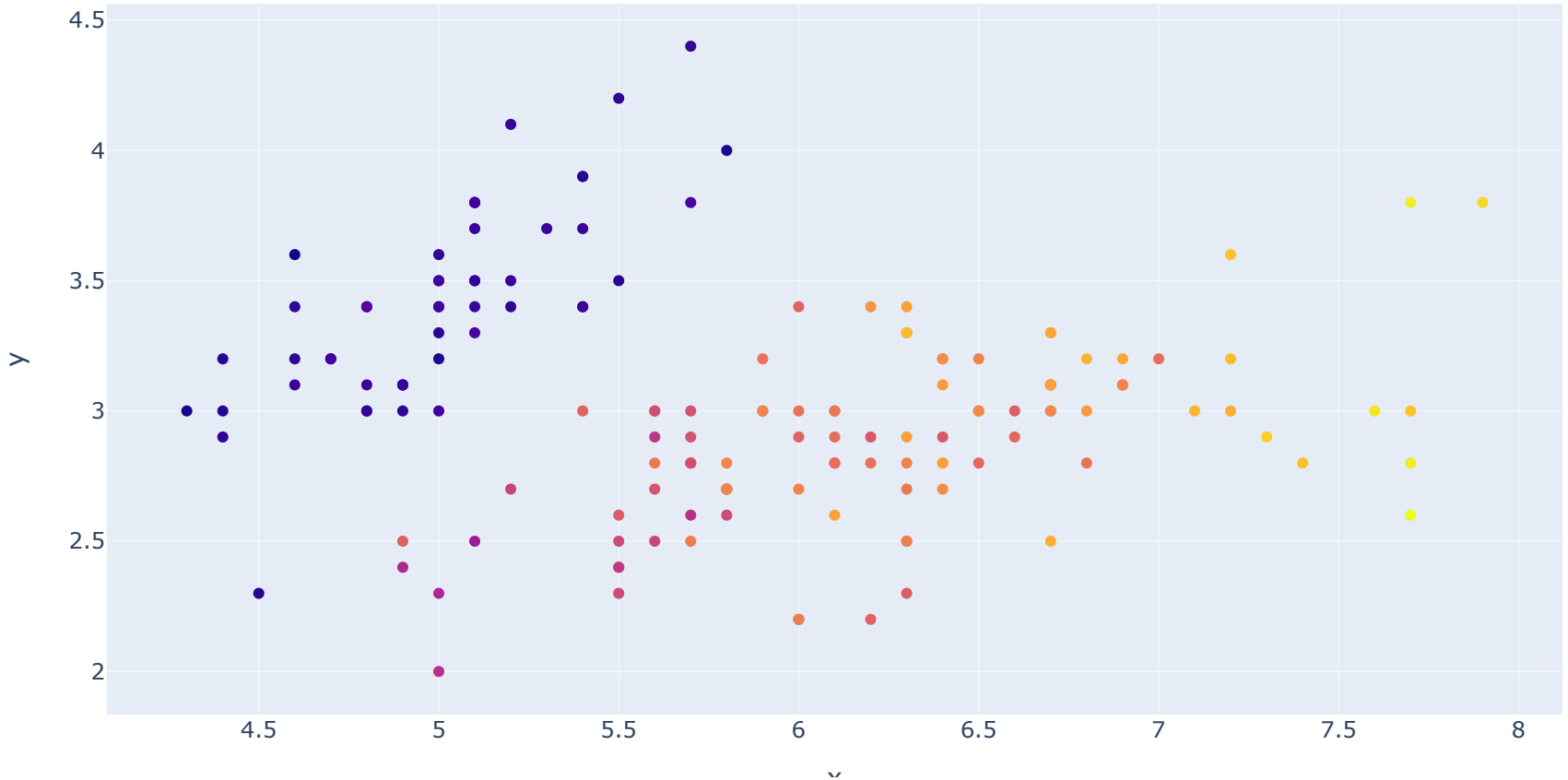
In [49]:
```
fig = px.scatter(x=df['PetalLengthCm'], y=df['PetalWidthCm'], color=df["SepalLengthCm"])
fig.show()
```



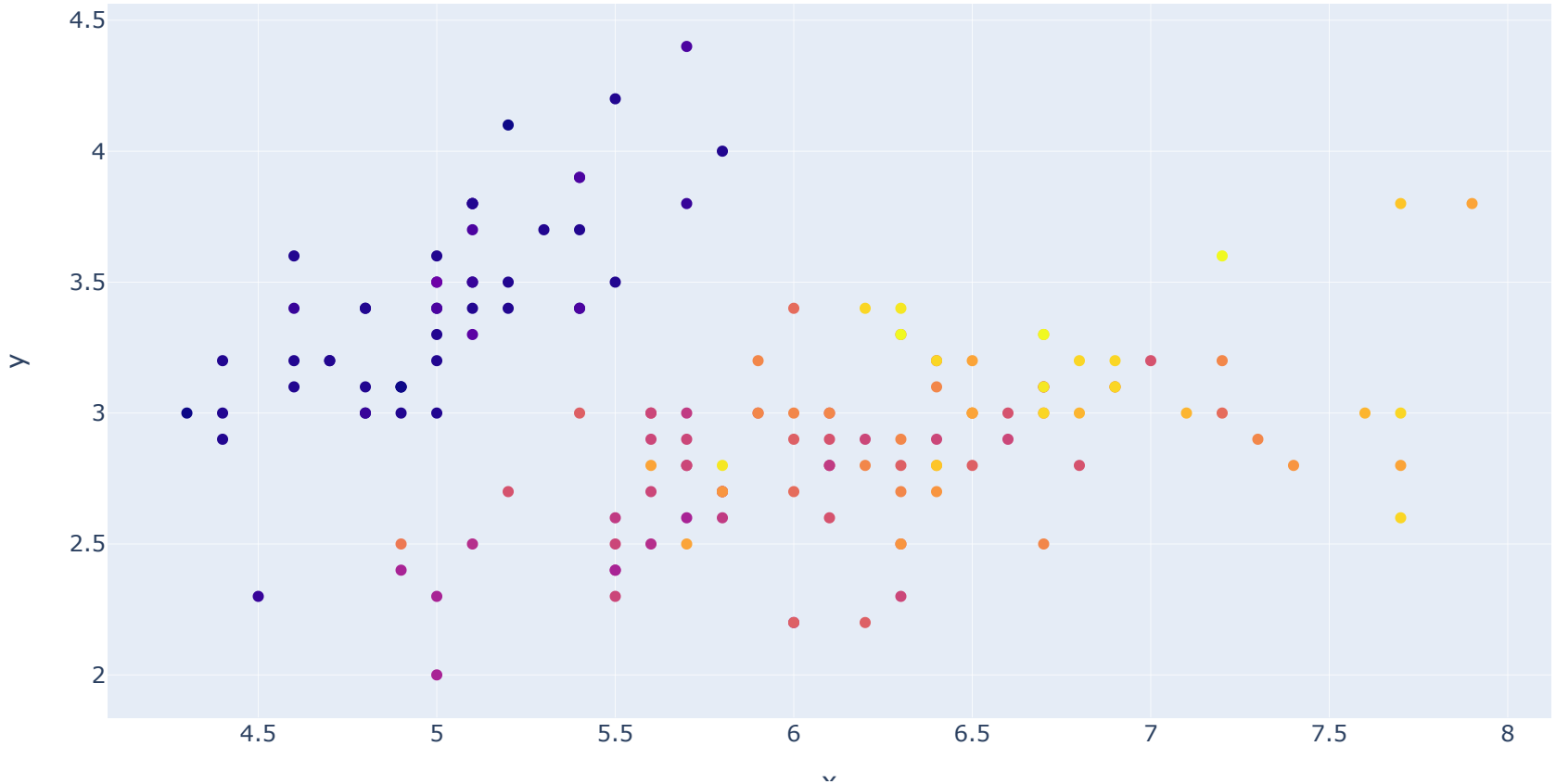
In [50]:
```
fig = px.scatter(x=df['PetalLengthCm'], y=df['PetalWidthCm'], color=df["SepalWidthCm"])
fig.show()
```



In [51]:
```
fig = px.scatter(x=df['SepalLengthCm'], y=df['SepalWidthCm'], color=df["PetalLengthCm"])
fig.show()
```



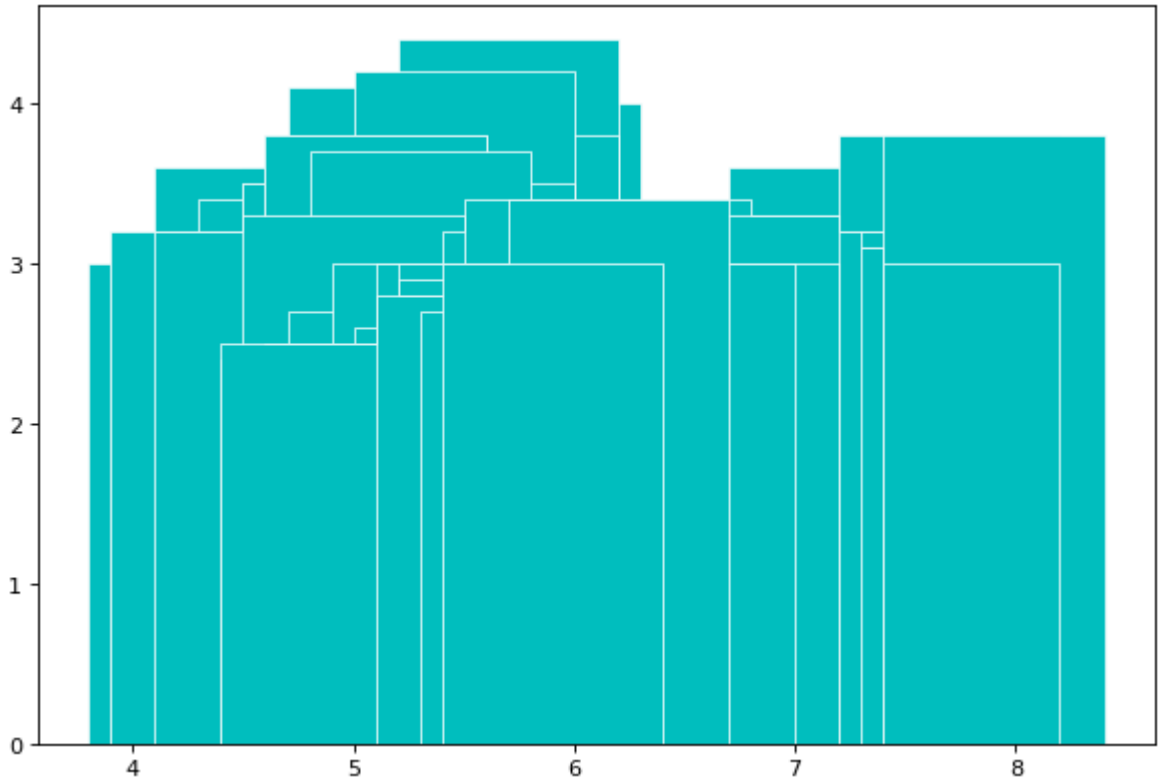
In [52]:
```
fig = px.scatter(x=df['SepalLengthCm'], y=df['SepalWidthCm'], color=df["PetalWidthCm"])
fig.show()
```



In [53]:
```
#Plotting Bar Plot for SepalLength, SepalWidth, PetalLength, PetalWidth
```

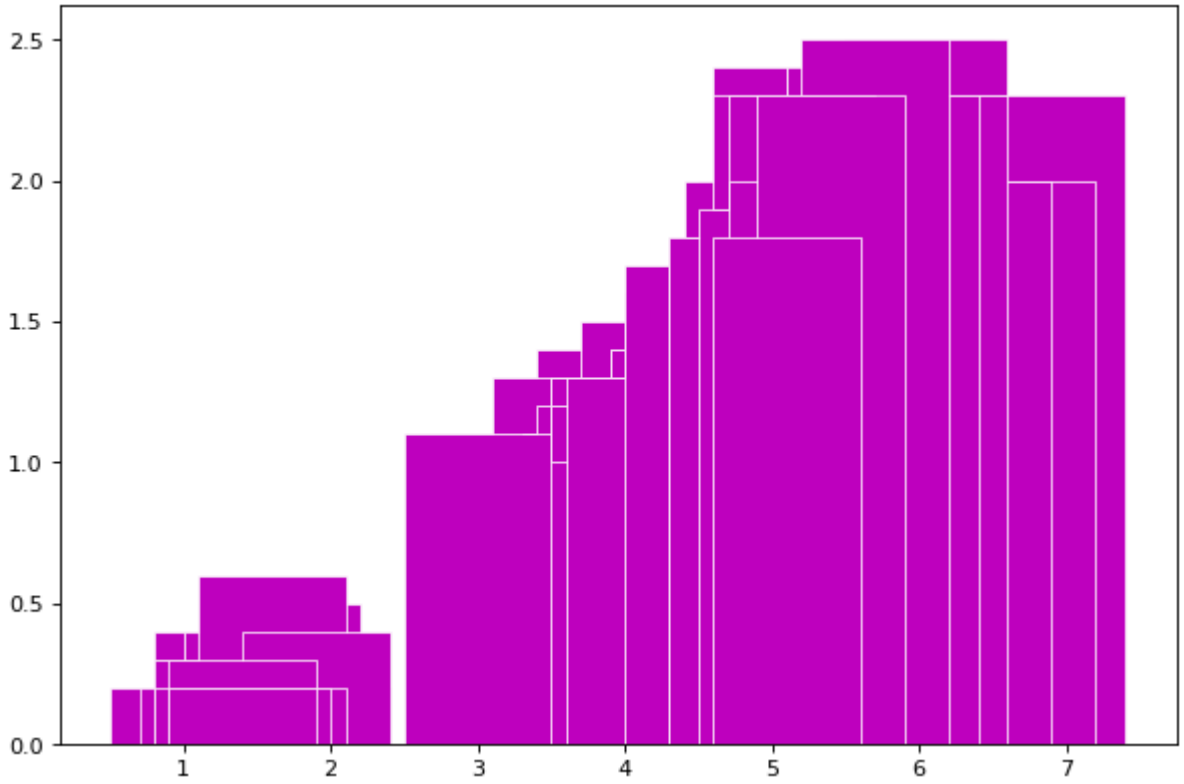
In [54]:
```
plt.bar(df['SepalLengthCm'], df['SepalWidthCm'], width=1, edgecolor="white", linewidth=0.7, color='c')
```

Out[54]: <BarContainer object of 150 artists>



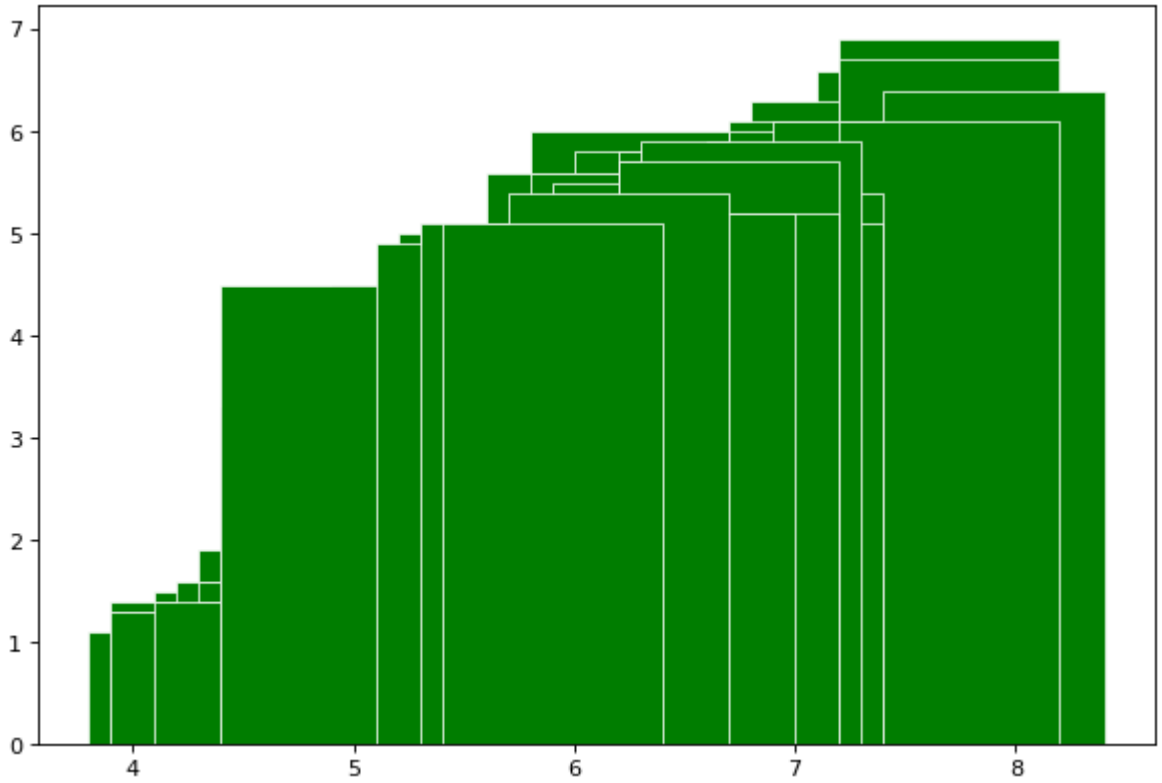
In [55]:
```
plt.bar(df['PetalLengthCm'], df['PetalWidthCm'], width=1, edgecolor="white", linewidth=0.7, color='m')
```

Out[55]:
```
<BarContainer object of 150 artists>
```



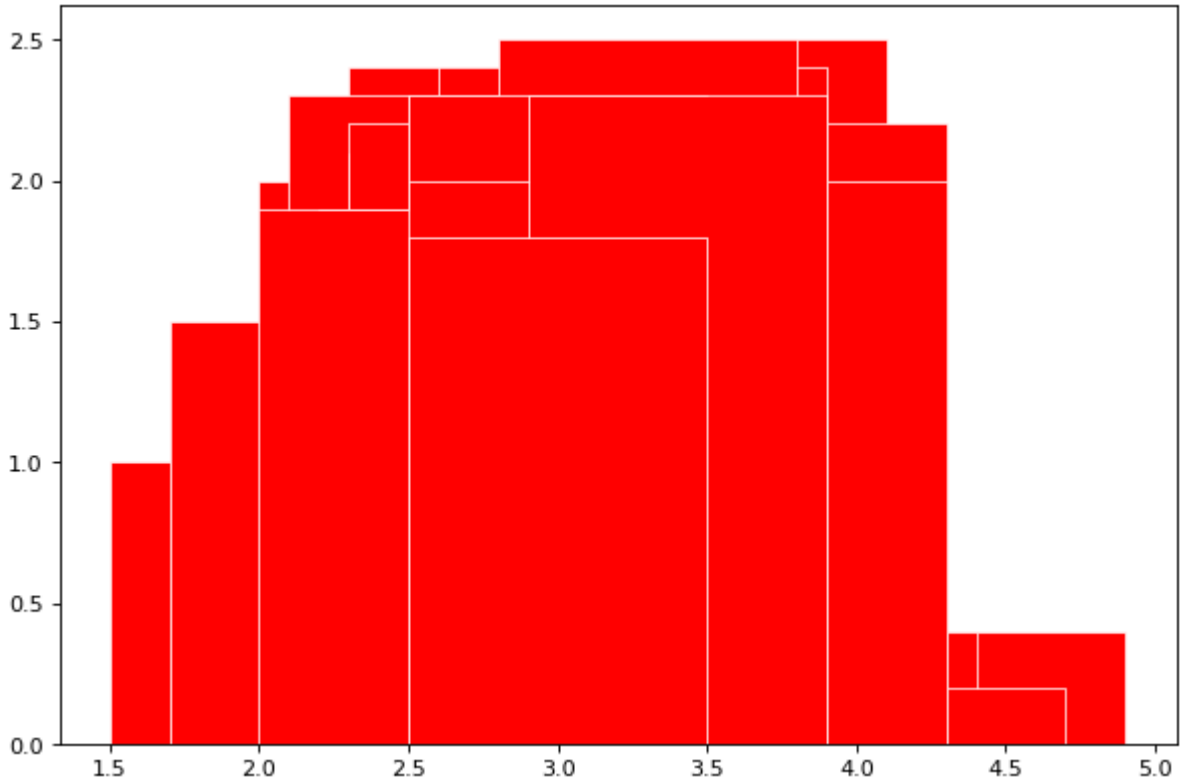
In [56]:
```
plt.bar(df['SepalLengthCm'], df['PetalLengthCm'], width=1, edgecolor="white", linewidth=0.7, color='g')
```

Out[56]:
```
<BarContainer object of 150 artists>
```



In [57]:
```
plt.bar(df['SepalWidthCm'], df['PetalWidthCm'], width=1, edgecolor="white", linewidth=0.7, color='r')
```

Out[57]:
```
<BarContainer object of 150 artists>
```



In [58]:
```
#calculating all possible statistical measures (mean, median, mode, standard deviation, etc.) for SepalLength colu
```

In [59]:
```
df1 = df['SepalLengthCm']
df1
```

Out[59]:
```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: SepalLengthCm, Length: 150, dtype: float64
```

In [60]:
```
df1.mean()
```

Out[60]:
```
5.843333333333334
```

In [61]:
```
df1.median()
```

Out[61]:
```
5.8
```

In [62]:
```
df1.mode()
```

Out[62]:
```
0    5.0
Name: SepalLengthCm, dtype: float64
```

In [63]:
```
df1.std()
```

Out[63]:
```
0.828066127977863
```

In [64]:
```
df1.skew()
```

Out[64]:
```
0.3149109566369728
```

In [65]:
```
df1.kurt()
```

Out[65]: -0.5520640413156395

In [66]:
```
df1.sum()
```

Out[66]: 876.5

In [67]:
```
df1.max()
```

Out[67]: 7.9

In [68]:
```
df1.min()
```

Out[68]: 4.3

In [69]:
```
df1.count()
```

Out[69]: 150

In [70]:
```
#calculating all possible statistical measures (mean, median, mode, standard deviation, etc.) for SepalWidth colum
```

In [71]:
```
df2 = df['SepalWidthCm']
df2
```

Out[71]:
```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: SepalWidthCm, Length: 150, dtype: float64
```

In [72]:
```
df2.mean()
```

Out[72]:
```
3.0540000000000003
```

In [73]:
```
df2.median()
```

Out[73]:
```
3.0
```

In [74]:
```
df2.mode()
```

Out[74]:
```
0    3.0
Name: SepalWidthCm, dtype: float64
```

In [75]:
```
df2.std()
```

Out[75]:
```
0.4335943113621737
```

In [76]:
```
df2.skew()
```

Out[76]:
```
0.3340526621720866
```

In [77]:
```
df2.kurt()
```

Out[77]: 0.2907810623654279

In [78]:
```
df2.sum()
```

Out[78]: 458.1

In [79]:
```
df2.max()
```

Out[79]: 4.4

In [80]:
```
df2.min()
```

Out[80]: 2.0

In [81]:
```
df2.count()
```

Out[81]: 150

In [82]:
```
#calculating all possible statistical measures (mean, median, mode, standard deviation, etc.) for PetalLength colu
```

In [83]:
```
df3 = df['PetalLengthCm']
df3
```

Out[83]:
```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: PetalLengthCm, Length: 150, dtype: float64
```

In [84]:
```
df3.mean()
```

Out[84]:
```
3.758666666666666
```

In [85]:
```
df3.median()
```

Out[85]:
```
4.35
```

In [86]:
```
df3.mode()
```

Out[86]:
```
0    1.5
Name: PetalLengthCm, dtype: float64
```

In [87]:
```
df3.std()
```

Out[87]:
```
1.7644204199522626
```

In [88]:
```
df3.skew()
```

Out[88]:
```
-0.27446425247378287
```

In [89]:
```
df3.kurt()
```

Out[89]: -1.4019208006454036

In [90]:
```
df3.sum()
```

Out[90]: 563.8

In [91]:
```
df3.max()
```

Out[91]: 6.9

In [92]:
```
df3.min()
```

Out[92]: 1.0

In [93]:
```
df3.count()
```

Out[93]: 150

In [94]:
```
#calculating all possible statistical measures (mean, median, mode, standard deviation, etc.) for PetalWidth colum
```

In [95]:
```
df4 = df['PetalWidthCm']
df4
```

Out[95]:
```
0      0.2
1      0.2
2      0.2
3      0.2
4      0.2
      ...
145    2.3
146    1.9
147    2.0
148    2.3
149    1.8
Name: PetalWidthCm, Length: 150, dtype: float64
```

In [96]:
```
df4.mean()
```

Out[96]:
```
1.1986666666666668
```

In [97]:
```
df4.median()
```

Out[97]:
```
1.3
```

In [98]:
```
df4.mode()
```

Out[98]:
```
0    0.2
Name: PetalWidthCm, dtype: float64
```

In [99]:
```
df4.std()
```

Out[99]:
```
0.7631607417008411
```

In [100]:
```
df4.skew()
```

Out[100]:
```
-0.10499656214412734
```

In [101]:
```
df4.kurt()
```

Out[101]: -1.3397541711393433

In [102]:
```
df4.sum()
```

Out[102]: 179.8

In [103]:
```
df4.max()
```

Out[103]: 2.5

In [104]:
```
df4.min()
```

Out[104]: 0.1

In [105]:
```
df4.count()
```

Out[105]: 150